



MoteView Users Manual

Revision A, May 2007

PN: 7430-0008-05



© 2003-2007 Crossbow Technology, Inc. All rights reserved.

Information in this document is subject to change without notice.

Crossbow, MoteWorks, IRIS, MICA, TrueMesh and XMesh are trademarks of Crossbow Technology, Inc. Other product and trade names are trademarks or registered trademarks of their respective holders

Table of Contents

1	Introduction.....	1
1.1	Wireless Mesh Networking Overview	1
1.2	MoteView Overview	2
1.3	Supported Sensor Boards and Mote Platforms	2
1.4	Supported Mote Software Applications	3
2	Installation.....	4
2.1	Supported PC Platforms and Operating Systems.....	4
2.2	Additional Software Requirements.....	4
2.3	Installation Steps.....	5
3	Application Quick Start	8
3.1	Verify PostgreSQL Installation.....	8
3.2	Connecting to a Live Sensor Network on your local PC	10
3.3	Connecting to a Live Sensor Network on a Remote PC or Stargate.....	12
3.4	Auto-discovery of Nodes	14
3.5	<i>MoteView</i> at a Glance	14
4	Node List.....	15
4.1	Add Node.....	16
4.2	Delete Node	16
4.3	Node Properties.....	17
5	Visualization Tabs	18
5.1	Data.....	18
5.2	Command.....	19
5.3	Chart	20
5.4	Health.....	22
5.5	Histogram	23
5.6	Scatterplot.....	24
5.7	Topology.....	24
5.8	Live/Historical/Playback Mode	29
6	Toolbar/Menus.....	30
6.1	File > Connect to WSN.....	30
6.2	File > Import	34
6.3	File > Export	34
6.4	File > Print Preview	35

6.5	File > Print	36
6.6	File> Exit	36
6.7	Settings > General.....	36
6.8	Settings > Node.....	36
6.9	Tools > Refresh View	36
6.10	Tools > Add Node	36
6.11	Tools > Program Mote	37
6.12	Tools > Alerts.....	37
6.13	Tools > MDA Support	39
6.14	Tools > Reset Database.....	43
6.15	Units	43
6.16	Help > User's Manual	43
6.17	Help > Support	43
6.18	Help > About.....	43
7	Server/Error/Status Messages	44
8	Server Administration.....	45
9	Database Administration	47
9.1	PostgreSQL.....	47
9.2	SQL.....	47
9.3	Database Tools.....	49
10	Health Packets.....	50
10.1	Node Health Packet Definitions	50

About This Document

The following annotations have been used to provide additional information.

NOTE

Note provides additional information about the topic.

EXAMPLE

Examples are given throughout the manual to help the reader understand the terminology.

IMPORTANT

This symbol defines items that have significant meaning to the user

WARNING

The user should pay particular attention to this symbol. It means there is a chance that physical harm could happen to either the person or the equipment.

The following paragraph heading formatting is used in this manual:

1 Heading 1

1.1 Heading 2

1.1.1 Heading 3

This document also uses different body text fonts (listed in Table 0-1) to help you distinguish between names of files, commands to be typed, and output coming from the computer.

Table 0-1. Font types used in this document.

Font Type	Usage
Courier New Normal	Sample code and screen output
Courier New Bold	Commands to be typed by the user
<i>Times New Roman Italic</i>	TinyOS files names, directory names
Franklin Medium Condensed	Text labels in GUIs

1 Introduction

1.1 Wireless Mesh Networking Overview

Wireless sensor networks have attracted a wide interest from industry due to their diversity of applications. A key to realizing their potential is multi-hop mesh networking which enables scalability and reliability.

A mesh network is a generic name for a class of networked embedded systems that share several characteristics including:

- **Multi-Hop**—the capability of sending messages peer-to-peer to a base station, thereby enabling scalable range extension;
- **Self-Configuring**—capable of network formation without human intervention;
- **Self-Healing**—capable of adding and removing network nodes automatically without having to reset the network; and
- **Dynamic Routing**—capable of adaptively determining the route based on dynamic network conditions (e.g., link quality, hop-count, gradient, or other metric).

When combined with battery power management, these characteristics allow sensor networks to be long-lived, easily deployed, and resilient to the unpredictable wireless channel. With mesh networking, the vision of pervasive and fine-grained sensing becomes reality.

A wireless network deployment is composed of the three distinct software tiers:

- The **Client Tier** provides the user visualization software and graphical interface for managing the network. Crossbow provides free client software called *MoteView* that bundles software from all three tiers to provide an end-to-end solution.
- The **Server Tier** is an always-on facility that handles translation and buffering of data from the wireless network and provides the bridge between the wireless motes and the internet clients. *XServe* and *XOtap* are server tier applications that can run on a PC or Stargate.
- The **Mote Tier**, where *XMesh* resides, is the software that runs on the cloud of sensor nodes forming a mesh network. The *XMesh* software provides the networking algorithms required to form a reliable communication backbone connecting all nodes within the mesh cloud to the server.

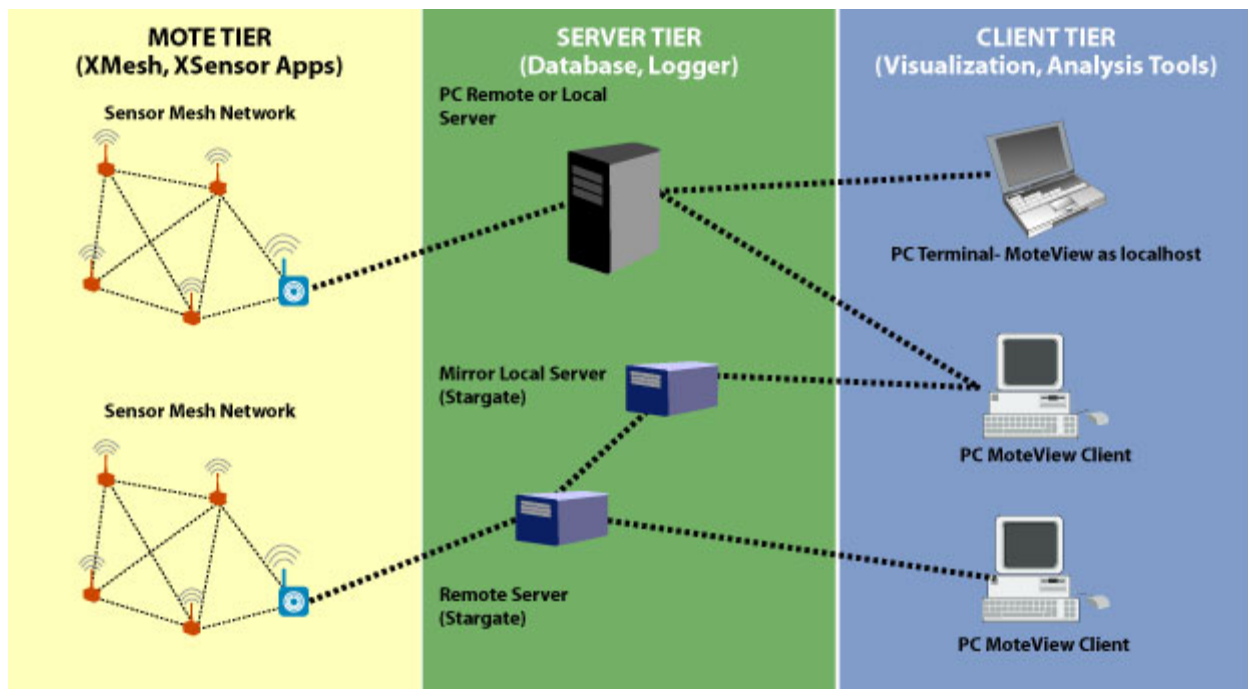


Figure 1-1. Software framework for a wireless sensor network

1.2 MoteView Overview

MoteView is designed to be an interface (“client tier”) between a user and a deployed network of wireless sensors. *MoteView* provides the tools to simplify deployment and monitoring. It also makes it easy to connect to a database, to analyze, and to graph sensor readings.

Figure 1-1 depicts a three-part framework for deploying a sensor network system. The first part is the Mote layer or sensor mesh network. The Motes are programmed with *XMesh*/TinyOS firmware (“application”) to do a specific task: e.g., microclimate monitoring, asset tracking, intrusion detection, etc. The second layer or Server tier provides data logging and database services. At this layer sensor readings arrive at the base station (e.g., MIB510, MIB520, MIB600, or Stargate) and are stored on a server or Stargate. The third part is the client tier in which software tools provide visualization, monitoring, and analysis tools to display and interpret sensor data. The purpose of this document is to explain the features of *MoteView* and to provide information on the supported Mote layer applications, Mote platforms, and sensor boards.

1.3 Supported Sensor Boards and Mote Platforms

MoteView supports all of Crossbow’s sensor and data acquisition boards (Table 1-1) as well as the MICA2, MICA2DOT, and MICAz processor/radio platforms (Table 1-2). In addition, *MoteView* can be used to deploy and monitor sensor integrated platforms such as the MSP Mote security/intrusion detection system and the MEP Mote environmental monitoring system (Table 1-3).

Table 1-1. Sensor (MTS series) and data acquisition boards supported by MoteView and their plug-and-play compatible Mote platforms

Sensor and Data Acquisition Boards	Mote Platforms			
	IRIS	MICAz	MICA2	MICA2DOT
MTS101		✓	✓	
MTS300/310	✓	✓	✓	
MTS410		✓		
MTS400/MTS420	✓	✓	✓	
MTS450		✓	✓	
MTS510				✓
MDA100	✓	✓	✓	
XBW-DA100		✓		
MDA300	✓	✓	✓	
MDA320	✓	✓	✓	
XBW-DA325		✓		
MDA500				✓

Table 1-2. Mote processor/radio (MPR) platforms supported by MoteView

Mote Platforms	Model Number(s)	RF Frequency Band(s)
IRIS	XM2110	2400 MHz to 2483.5 MHz
	M2110	2400 MHz to 2483.5 MHz
MICAz	MPR2400 MPR2600	2400 MHz to 2483.5 MHz 2400 MHz to 2483.5 MHz
MICA2	MPR400 MPR410 MPR420 MPR600	868 MHz to 870 MHz; 903 MHz to 928 MHz 433.05 to 434.8 MHz 315 MHz (for Japan only) 868 MHz to 870 MHz; 903 MHz to 928 MHz
MICA2DOT	MPR510 MPR510 MPR520	868 MHz to 870 MHz; 903 MHz to 928 MHz 433.05 to 434.8 MHz 315 MHz (for Japan only)

Table 1-3. Sensor integrated (MEP, MSP) platforms supported by MoteView

Sensor Integrated Mote Platforms	Description of Usage
MEP410	Microclimate and ambient light monitoring
MEP510	Temperature and humidity monitoring
MSP410	Physical security and intrusion detection

1.4 Supported Mote Software Applications

XMesh is Crossbow's multi-hop mesh networking protocol that has various options including low-power listening, time synchronization, sleep modes, any-to-base and base-to-any routing. All of our sensor and data acquisition boards are supported with *XMesh* enabled applications. Refer to the *MoteConfig User's Manual* for details.

2 Installation

2.1 Supported PC Platforms and Operating Systems

MoteView is supported on the following platforms:

- *Windows XP Home*
- *Window XP Professional*
- *Windows 2000 with SP4*

❗ **NOTE:** The filing system on the hard disk must be NTFS.

For information on converting from FAT to NTFS, please click on the following link:

<http://www.microsoft.com/technet/prodtechnol/winxpro/maintain/convertfat.mspx>

2.1.1 PC Interface Port Requirements

The gateway platform used in the base station determines the PC interface port required by *MoteView*:

1. For a **MIB510** serial gateway: an RS-232 serial port.
2. For a **MIB520** USB gateway: a USB port.
3. For a **MIB600** Ethernet gateway: A wired Ethernet or 802.11 wireless card only if the MIB600 is on a LAN with wireless access.
4. For a **Stargate** or other server: A wired Ethernet, an 802.11 wireless card only if the Stargate has a wireless modem or is on a LAN with wireless access, or a cellular modem for wireless Internet access.

2.2 Additional Software Requirements

For the application to run the following additional components are required:

- PostgreSQL 8.0 *database service*
- PostgreSQL *ODBC driver*
- Microsoft .NET 1.1 *framework*

The installation files for these components are included on the *MoteView* installation CD.

2.2.1 PostgreSQL

All the visualization tools in *MoteView* require being connected to a database. This database can reside on your PC ("localhost"), a remote server, or a Stargate. The size of this database is bound by the storage available on the system.

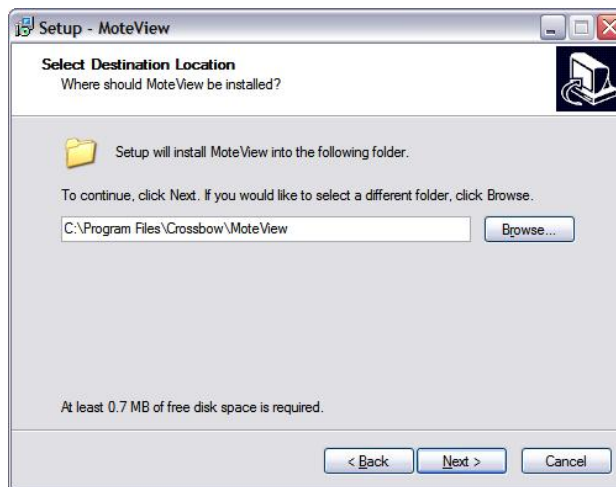
❗ **IMPORTANT:** The *PostgreSQL 8.0* installation requires administrative privileges on the system, including the ability to create a new user called *postgres* (for *PostgreSQL*).

IMPORTANT: Running the *PostgreSQL* database service is required if your PC is hosting both the server and client layer functions.

2.3 Installation Steps

Shut down all the programs running on your computer.

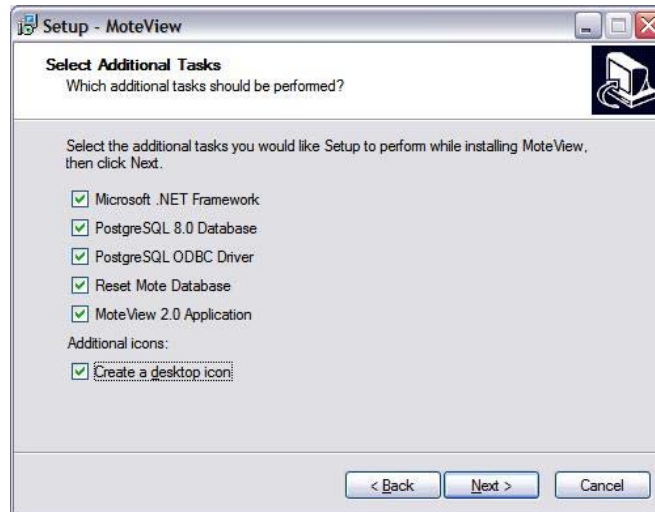
1. Insert the *MoteView Support* CD into the computer's CD drive.
2. Double-click on ***MoteViewSetup.exe*** from *MoteView* folder.
3. At the **Welcome to the MoteView Setup Wizard** window, click **Next>**.
4. Select the desired installation directory, then click **Next>**.



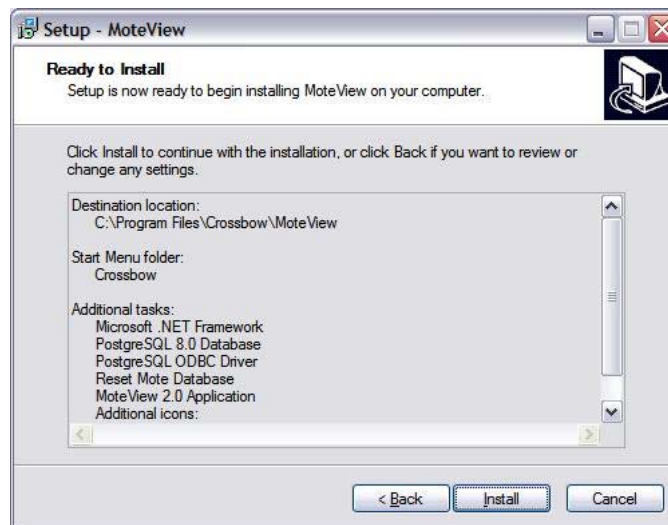
5. Select the desired Start Menu folder name and click **Next>**.



6. Select all available installation tasks and click **Next>**.

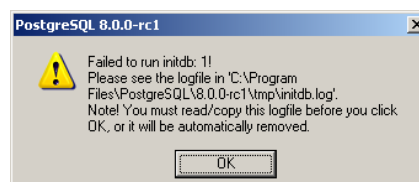


7. Confirm your selections and click **Install**.



Execute the .NET framework and *PostgreSQL* ODBC installation prompts if applicable and follow the instructions.

NOTE: If you have a version of MoteView prior to v1.0 you may get a *PostgreSQL 8.0.0-rc1* window when installing the database as shown below.



Record the exact error message in *C:\Program Files\PostgreSQL\8.0.0-rc1\tmp\initdb.log*. In most cases you can simply click the OK button and proceed. However, in other cases you may have to do the following:

- a. Uninstall *PostgreSQL 8.0* from **Start>Control Panel>Add or Remove Programs**

- b. Remove the *C:\Program Files\PostgreSQL* directory manually through the Windows Explorer utility (Start>Right Click>Explore).
- c. Reinstall *MoteView* with all checked **except** the .NET Framework.

8. Please note that if you receive an MDAC warning such as the one pictured below, you may ignore this warning and continue on with the installation. The warning is simply notification that your operating system contains an MDAC version that is newer than what *MoteView* is installing. This warning can be ignored by clicking on **Cancel**.




9. When the Setup Wizard has finished it will ask if you want to start *MoteView*. You may start *MoteView* now, but in some cases it may ask you to restart your computer first.
10. If you have restarted or at any time after you can start *MoteView* by double-clicking on the *MoteView* icon or by going to the installation folder (the default install folder is *C:\Program Files\Crossbow\MoteView*) and double-clicking on *MoteView.exe*.

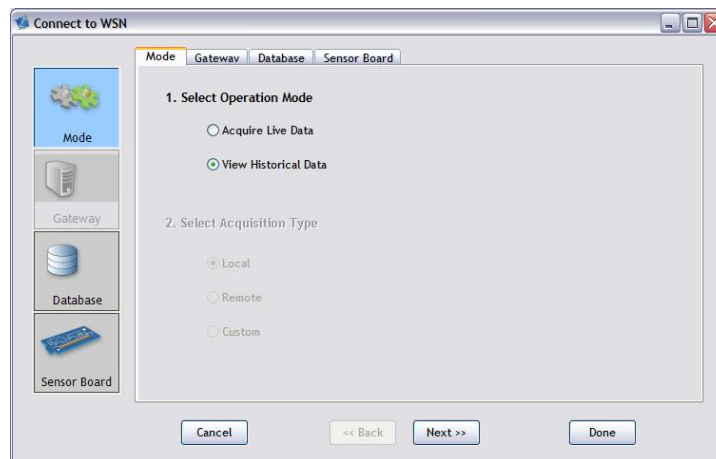
3 Application Quick Start

Once a sensor network is running and *MoteView* is installed on a PC, minimal configuration is necessary to start collecting data from the sensor network.

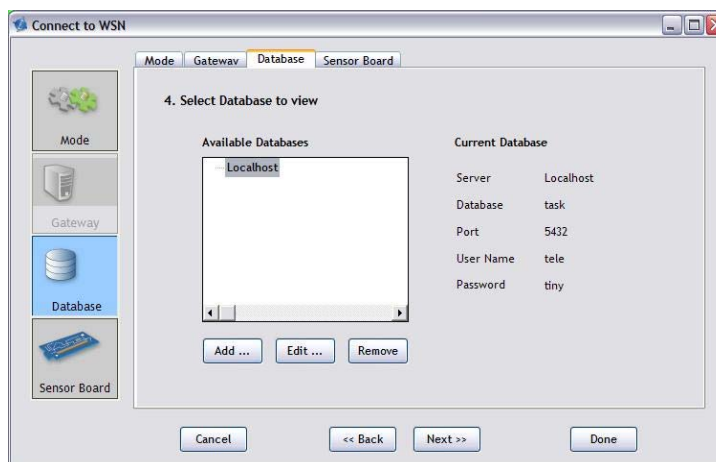
3.1 Verify PostgreSQL Installation

During the installation of *MoteView* a static database was included to make it possible to demonstrate *MoteView*'s features without having to be connected to an active sensor network or a remote server/database. The steps described here also apply to viewing data collected from an active sensor network.

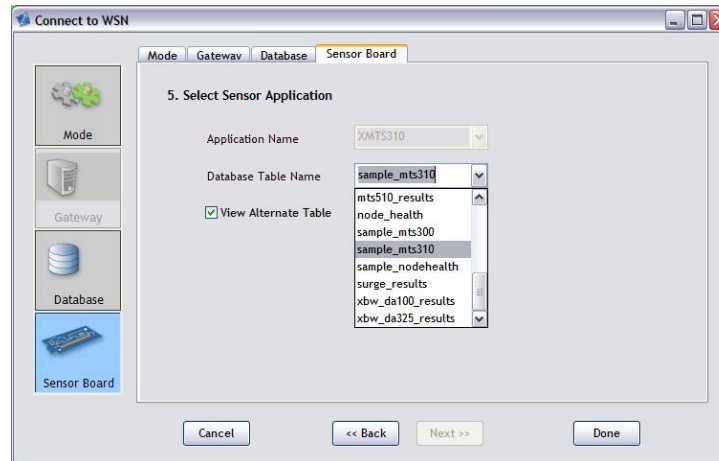
1. Click on the Connect to WSN icon , in the upper left hand corner or select **File > Connect to WSN...** from the menubar. The **Connect to WSN Wizard** will appear.



2. Check on **View Historical Data** in the **Mode** tab and click on **Next >>**.
3. Make sure that the **Database** tab shows the current database highlighted as `localhost` and click on **Next >>**.



4. In the **Sensor Board** tab, check on **View Alternate Table** and select `sample_mts310` table from the **Database Table Name** dropdowns and click on **Done**.

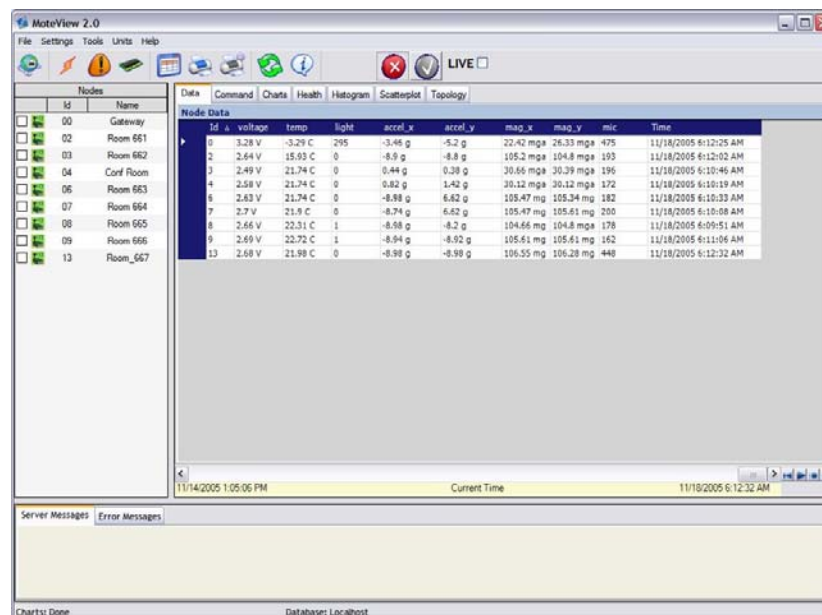


❗ **NOTE:** If the task database is not available, or if the table names are empty, the PostgreSQL install was not successful. Verify that the *psqlODBC* drivers have been installed using the **Control Panels>Add or Remove Programs** wizard. Verify that the postgres database service has started using the *Task Manager*. If not, use the **Control Panels>Administrative Tools>Services** to start the postgres service.

A good source of debugging information is

http://pginstaller.projects.postgresql.org/faq/FAQ_windows.html

5. *MoteView* will then access the `sample_mts310` table in the database task and show Data view as shown below.



❗ **IMPORTANT:** There are two ways to repair a corrupt database installation:

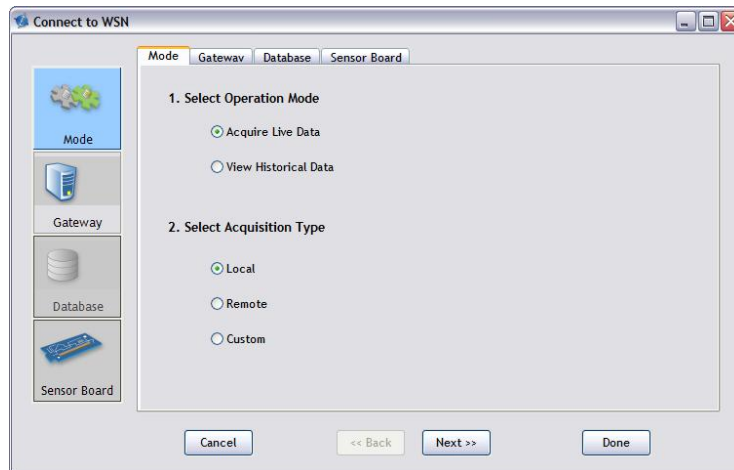
1. Select **Tools > Reset Database** from the MoteView menu
2. Open a File Manager, navigate to `C:\Program Files\Crossbow\MoteView`, and execute "resetdb.cmd".

A *sample_mts300* table and a *sample_nodehealth* table are also included in the default *task* database created by the *MoteView* installer.

3.2 Connecting to a Live Sensor Network on your local PC

Use the following steps to access data from a live sensor network connected to your local PC via the MIB510, MIB520 or MIB600 gateway.

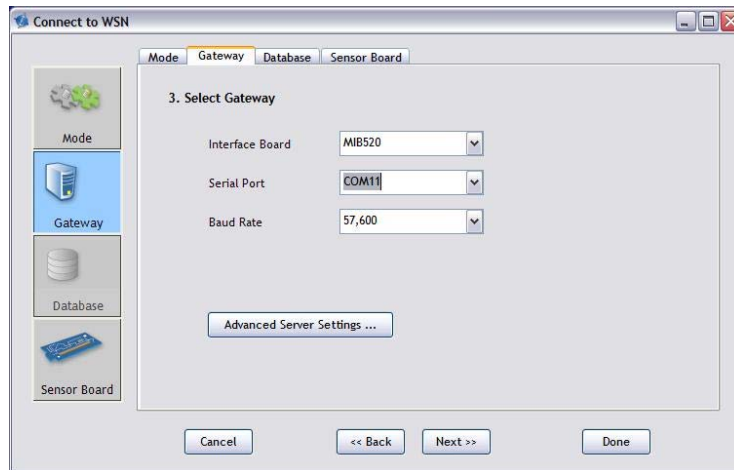
1. Select **File > Connect to WSN...** from the menu. Select the **Mode** tab, check on **Acquire Live Data** as operation mode and **Local** as acquisition type and click on **Next >>**.



2. In the **Gateway** tab, specify the Interface Board type, Port/Host Name etc as described below.
 - i. If using a MIB510, in the **Gateway** tab make sure that the MIB510's COM is set to the correct port number and that the **baud rate** is **57600**.



- ii. If using MIB520, enter the higher of the 2 COM ports installed by the MIB520's driver and set the **baud rate** to **57600**.



❖ **NOTE:** The **MIB520** requires the installation of the *FTDI FT2232C* drivers. Once these drivers are installed, the **Device Manager** (**Start > Control Panel > System > Hardware**) will display the **MIB520** as two new virtual com ports. Refer to the *MPR-MIB Series User's Manual* for details.

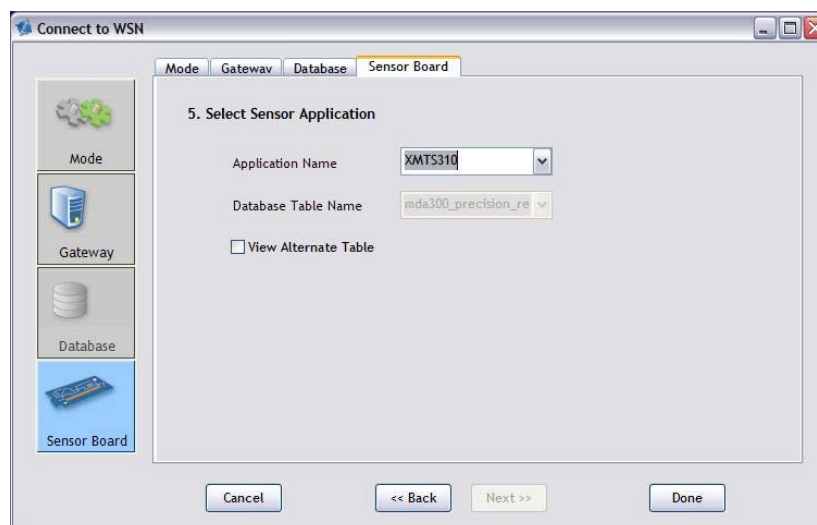
- iii. If using a **MIB600**, select **MIB600** from Interface Board dropdown and enter the IP address of the MIB600 in the **Hostname** text-box. The **Port** should default to **10002**.



❖ **NOTE:** The IP address of the **MIB600** can be identified by using the *Lantronix DeviceInstaller* application. Refer to the *MPR-MIB Series User's Manual* for details.

Once the Gateway settings are complete, click on **Next >>**.

3. In the **Sensor Board** tab, uncheck the **View Alternate Table** checkbox and choose the **XMesh Application Name** that matches the firmware programmed into the Mote from **Application Name** dropdown. Click on **Done**.



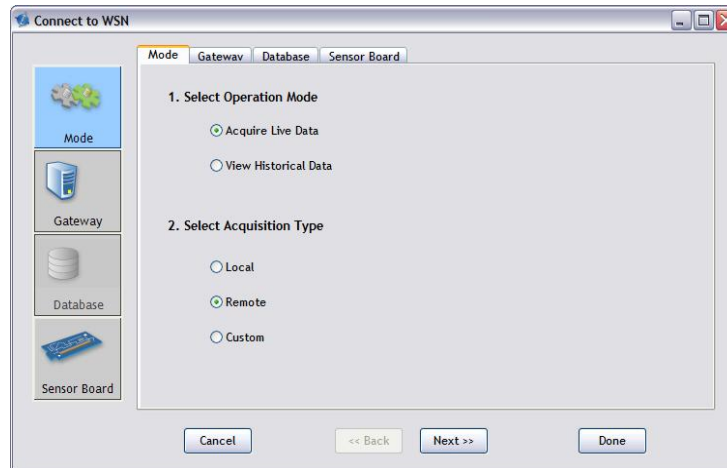
If you have an <i>XMesh</i> Application...	Then choose one of the following pull down choices
Default. Use when you don't see the <i>XMesh</i> application	(none)
<i>XMTS101_<freq>_<mode></i>	XMTS101
<i>XMTS300_<freq>_<mode></i>	XMTS300
<i>XMTS310_<freq>_<mode></i>	XMTS310
<i>XMTS400_<freq>_<mode></i>	XMTS400
<i>XMTS410_<freq>_<mode></i>	XMTS410
<i>XMTS420_<freq>_<mode></i>	XMTS420
<i>XMTS450_<freq>_<mode></i>	XMTS450
<i>XMTS510_<freq>_<mode></i>	XMTS510
<i>XMDA100_<freq>_<mode></i>	XMDA100
<i>XBW-DA100_<freq>_<mode></i>	XBW-DA100
<i>XMDA300_<freq>_<mode></i>	XMDA300
<i>XMDA300p_<freq>_<mode></i>	XMDA300P
<i>XMDA320_<freq>_<mode></i>	XMDA320
<i>XBW-DA325_<freq>_<mode></i>	XBW-DA325
<i>XMDA500_<freq>_<mode></i>	XMDA500
<i>XMEP410_<freq>_<mode></i>	XMEP-SYS
<i>XMEP510_<freq>_<mode></i>	XMEP-SYS
<i>XMSP410_<freq>_hp.exe</i>	XMSP410

- If you are not able to receive data, you may also need to check the **Live** check box on the main *MoteView* screen if it has not been previously checked. Use the **Server Messages** pane at the bottom of your *MoteView* display to verify that node data is being received by your PC.

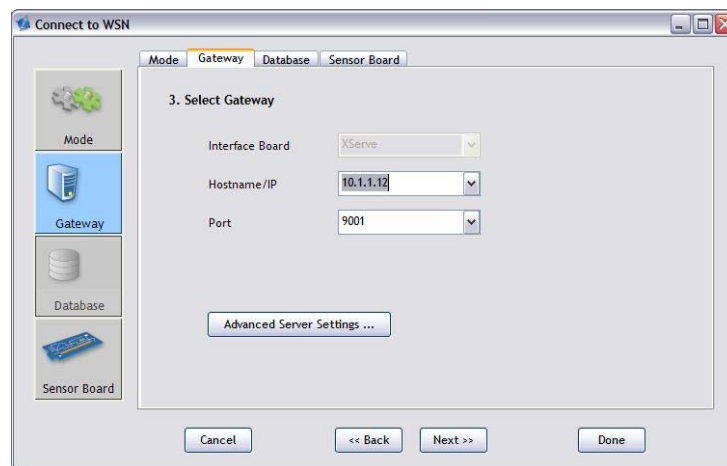
3.3 Connecting to a Live Sensor Network on a Remote PC or Stargate

To connect to a live sensor network on a remote PC requires changes to the database server and gateway settings.

1. Select **File > Connect to WSN...** from the menu. Select the **Mode** tab, check on **Acquire Live Data** as operation mode and **Remote** as acquisition type and click on **Next >>**.



2. In the **Gateway** tab, type the **Hostname/IP address** of the remote server/Stargate. The **Port** should default to **9001**. Click on **Next >>**.



❖ **NOTE:** The **Port**, **Username**, and **Password** fields are preset to reasonable defaults and should **not** have to be changed.

3. In the **Sensor Board** tab, choose the **XMESH Application Name** that matches the firmware programmed into the Mote from **Application Name** dropdown. Click on **Done**.

❖ **NOTE:** If you are unable to view the tables on a remote database, verify that the remote PC's firewall accepts connections from your client PC. Also, check that the `pg_hba.conf` file has valid authentication information, and that the `postgresql.conf` file has the following line

```
listen_addresses = '*'
```

3.4 Auto-discovery of Nodes

New nodes will also appear in the upper left hand corner of the **Topology** view. These nodes can be dragged to their correct position on the topology map and those placements can be saved to the configuration file in the database.

3.5 MoteView at a Glance

MoteView has four main user interface sections:

1. Toolbar / Menus: Allows the user to specify actions and initiate command dialogs.
2. Node List: Shows all known nodes in a deployment and health status summary.
3. Visualization Tabs: Enables the user to view the sensor data in various ways.
4. Server / Error Messages: Displays a log of server events and incoming messages.

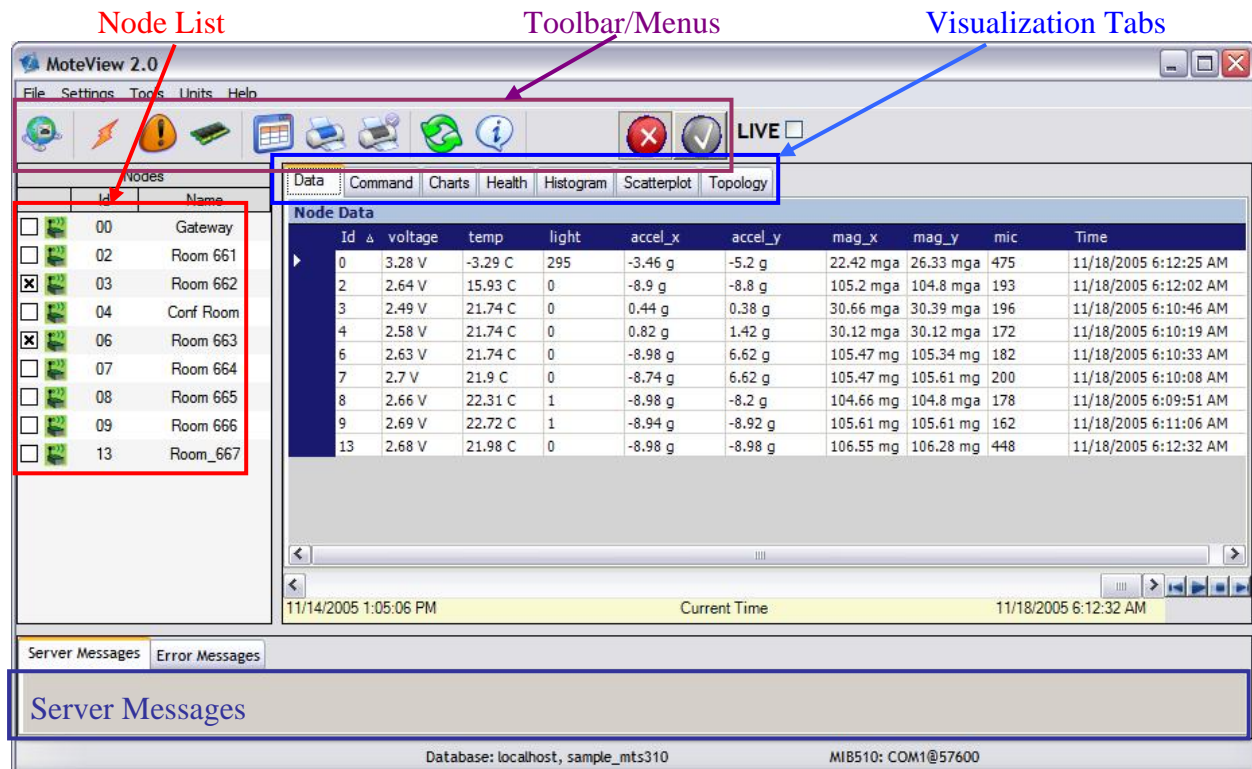


Figure 3-1. Screenshot of the MoteView GUI

More details about these user interfaces and their functional descriptions will be covered in the subsequent chapters.







4 Node List

The Node List displays all the known nodes in a deployment. This section allows users to configure the visualization settings for the nodes.

The users can sort the nodes by clicking on the column header.

The checkbox next to the node IS selects nodes to be plotted in Charts/Histogram/Scatterplot view.

Table 4-1. Icon properties of the node list

Node List Icon	Description
 Gray Mote icon	No results received
 Green Mote icon	Fresh results within the last 20 minutes
 Moss (light green) Mote icon	Results stale by >20 minutes
 Yellow Mote icon	Results stale by >40 minutes
 Orange Mote icon	Results stale by >60 minutes
 Red Mote icon	Results stale by more than a day

The time intervals for the node color changes can be edited from the dialog box that appears when you right-click on the node and select **Health Status Settings**.

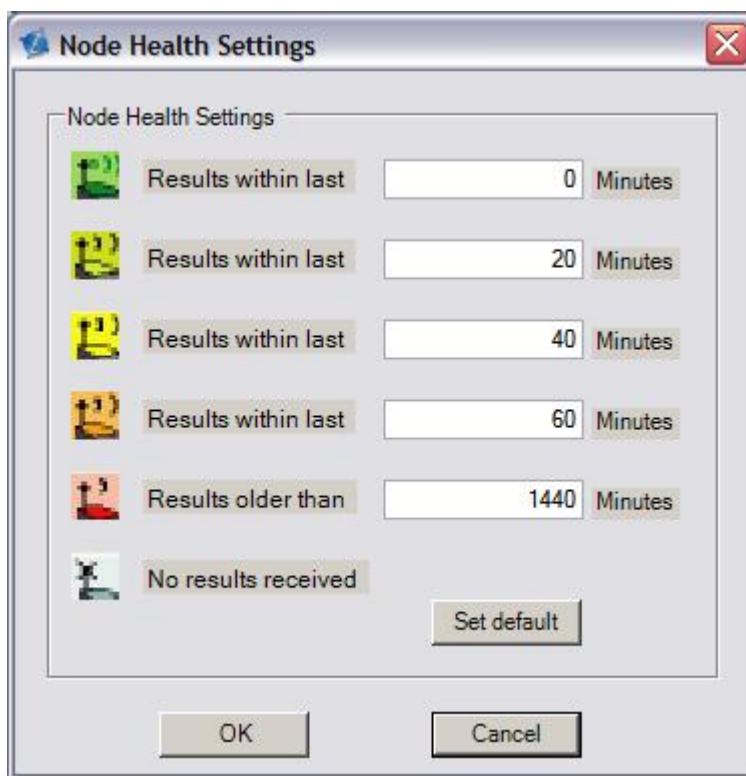


Figure 4-1. Screenshot of the Node Health Settings window.

4.1 Add Node

This dialog can be opened by right-clicking on the Node List and selecting **Add Node**. This will bring up the Mote Properties dialog for creation of a new node. The user can select a unique node ID and name the node at this time.

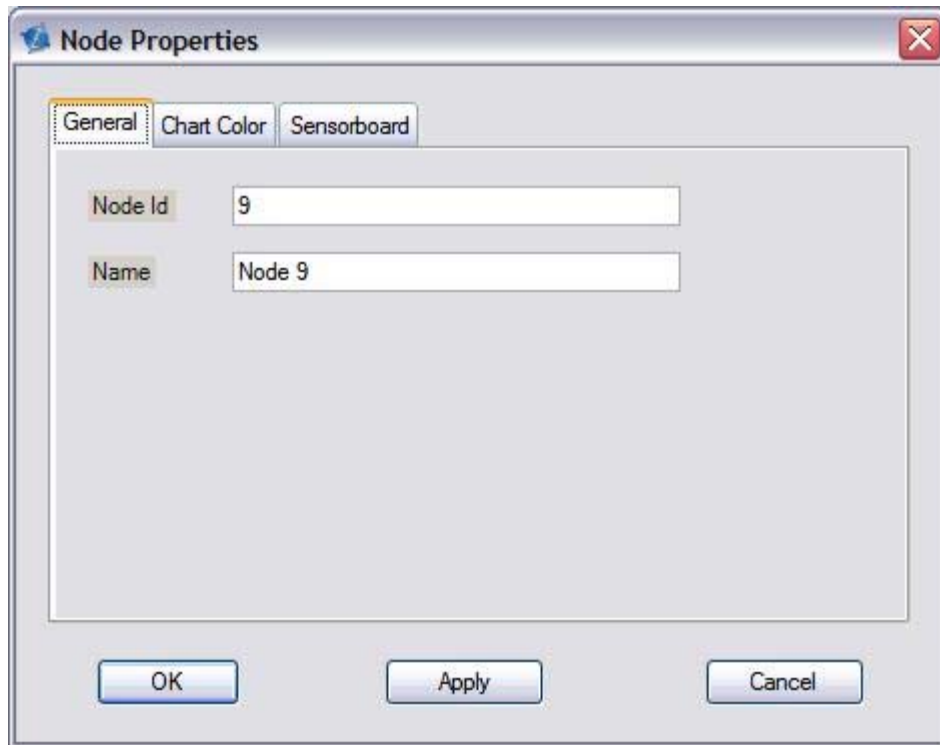


Figure 4-2. Screenshot of the Add Node dialog window.

4.2 Delete Node

Users can delete a node by highlighting a node, right-clicking on it and selecting **Delete Node**. A warning message would appear requesting confirmation of the delete operation.

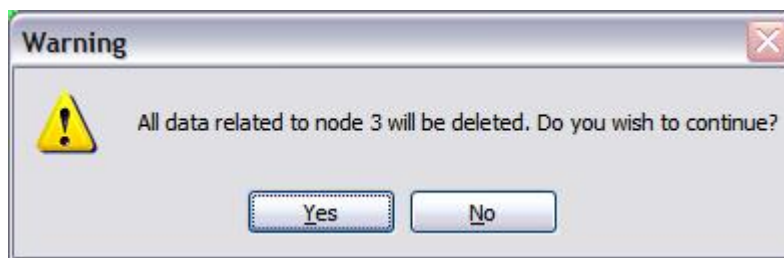


Figure 4-3. Screenshot of the Delete Node dialog window.

⚠️ WARNING

❗ **NOTE:** The Delete Node operation will permanently delete all the data associated with the specified node ID from the database.

4.3 Node Properties

This dialog is opened by selecting the **Properties** menu when right clicking a node in the Topology map, or by double-clicking a node in the node list. This allows the user to assign a **Name** to the Mote.

The **Color** tab allows the user to select the charting color for the Mote.

◀ **NOTE:** The **Sensorboard** tab of this dialog is only used to display calibration coefficients for boards that are equipped with an Intersema pressure temperature sensor such as the MTS400, MTS420, and MEP410.

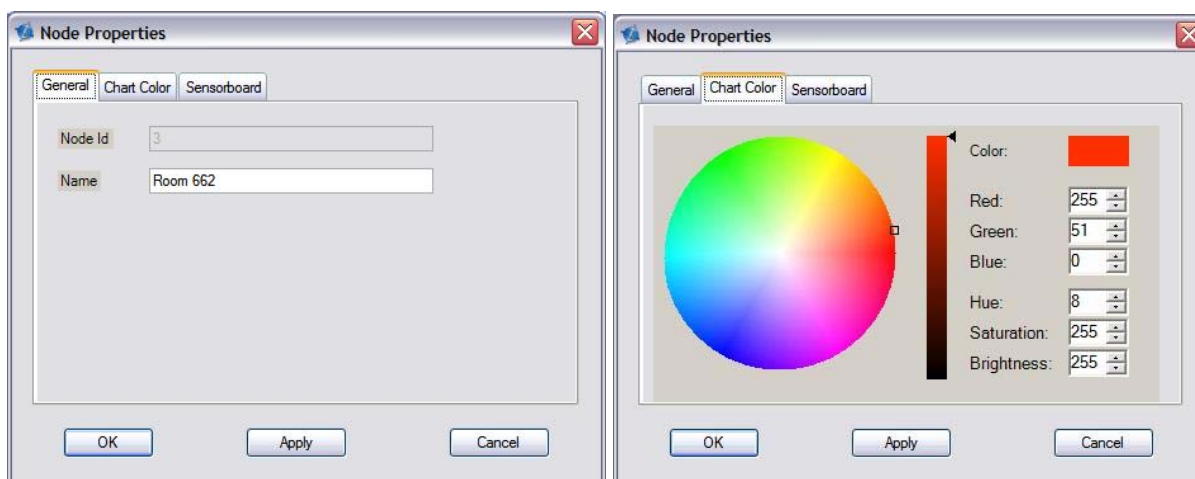


Figure 4-4. Screenshots of the Mote Properties dialog window.

5 Visualization Tabs

Seven visualization tabs (**Data**, **Command**, **Charts**, **Health**, **Histogram**, **Scatterplot** and **Topology**) provide different methods of viewing your sensor data.

5.1 Data

The **Data** tab displays the latest sensor readings received for each node in the network (see Figure 4-1 for an example from the demo database). The columns include node ID, server timestamp and sensor values from the sensor board firmware packet. The sensor data is automatically converted into standard engineering units.

Left-clicking the column header allows you to sort by node ID, parent, temperature, voltage, last result time, or any other sensor reading. Right-clicking the column header displays a pop-up menu with unit conversions relevant to the sensor.

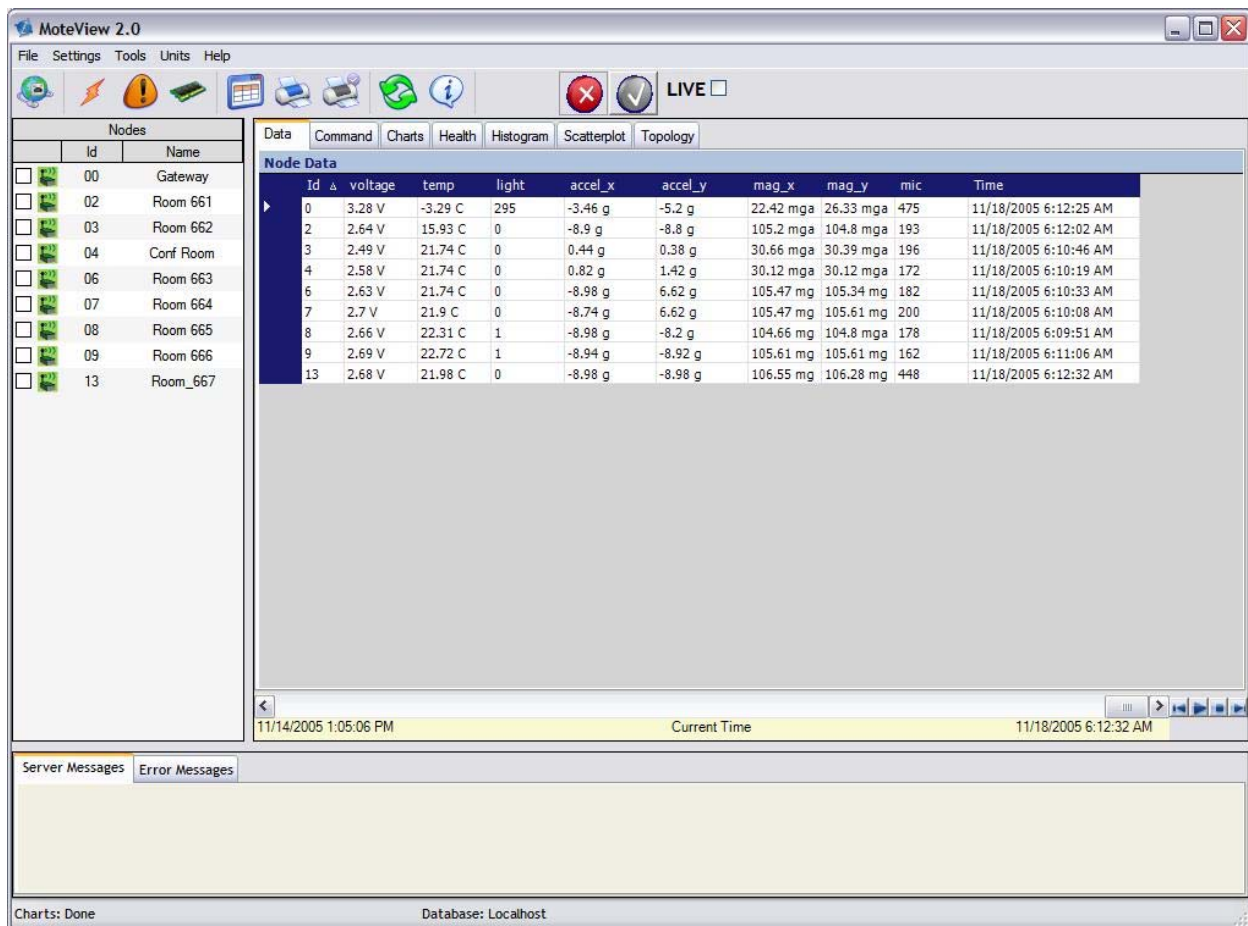


Figure 5-1. Screenshot of a demo database displayed in the **Data** tab

5.2 Command

The **Command** tab provides the user with an ability to change different node parameters wirelessly.

❏ **NOTE:** These commands do not get stored in the EEPROM and hence are volatile.

To modify the data sample and transmission rate, do the following:

1. Click on the **System** tab on the left side of the **Command** tab.
2. Select the **Node ID**. Check “**All Nodes**” to modify the data rate for all nodes in your network
3. Enter a data rate and press **Set**. The minimum data rate supported is 300 ms for high-power nodes. The default data rate for most high-power applications is 2 seconds. The default rate for low-power application is 3 minutes

To obtain the 64-bit unique ID of a Mote, do the following:

4. Click on the **System** tab on the left side of the **Command** tab.
5. Select the **Node ID**. Press **Get**

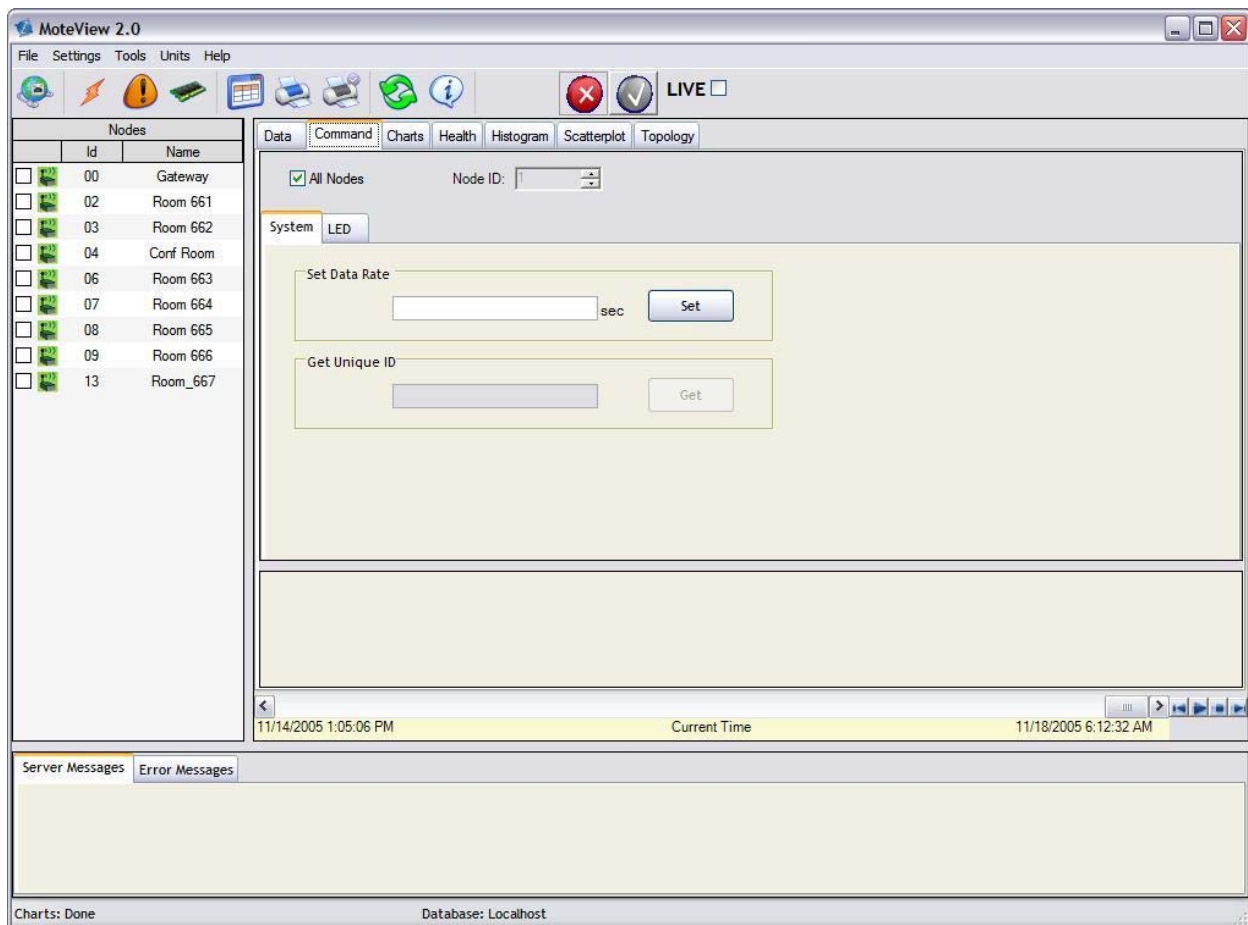


Figure 5-2. Screenshot of the System Configuration in the *Command Tab*.

To change the LED status, do the following.

1. Click the **LED** tab on the left side of the **Command** tab.
2. Select the **Node ID** of your network. If changing the system parameters for all nodes, check the **All Nodes** checkbox.
3. Select the Red, Yellow or Green LED(s) and check
ON – To turns the LED On.
OFF – To turns the LED Off
TOGGLE – To toggle the state of the LED
4. Press **SET** to send the specified commands to node (s) in your network.
5. Press **SET ALL** to activate all LEDs simultaneously.

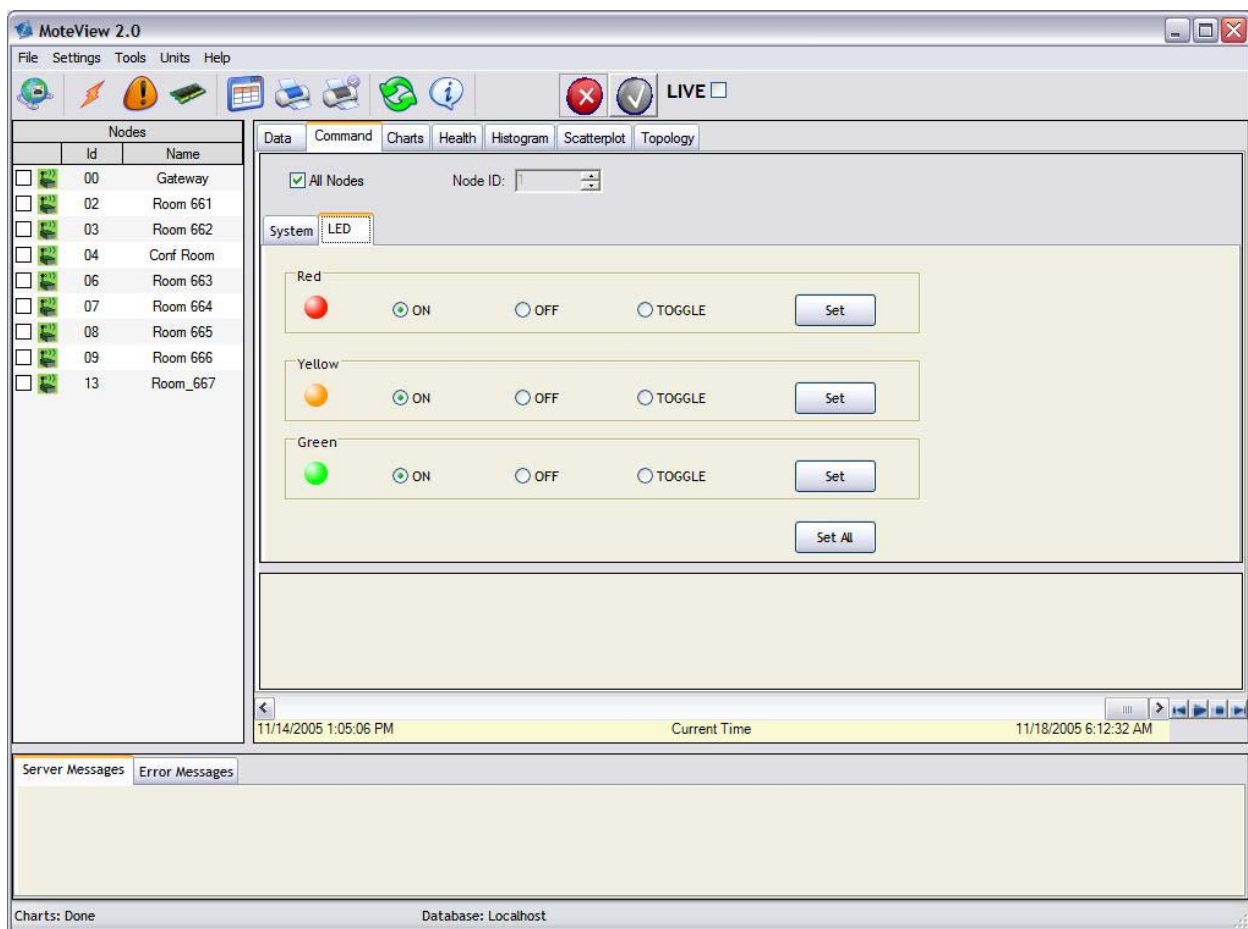


Figure 5-3. Screenshot of the LED actuation in the *Command Tab*

5.3 Chart

The **Chart** tab provides the ability to generate graphs of a sensor reading against time for a set of nodes. The following features and constraints apply to the graphs plotted in this view.

- Up to 3 sensor types can be selected for plotting, i.e., 3 different graphs.

- Up to 24 different Nodes can be selected for plotting by checking on the boxes next to the node list on the left side. A different plot color will be used for each node; a legend is displayed on the right side of the window.
- The x-axis on the graph shows the date and time.
- The y-axis on the graph shows data in engineering units for the sensor readings.

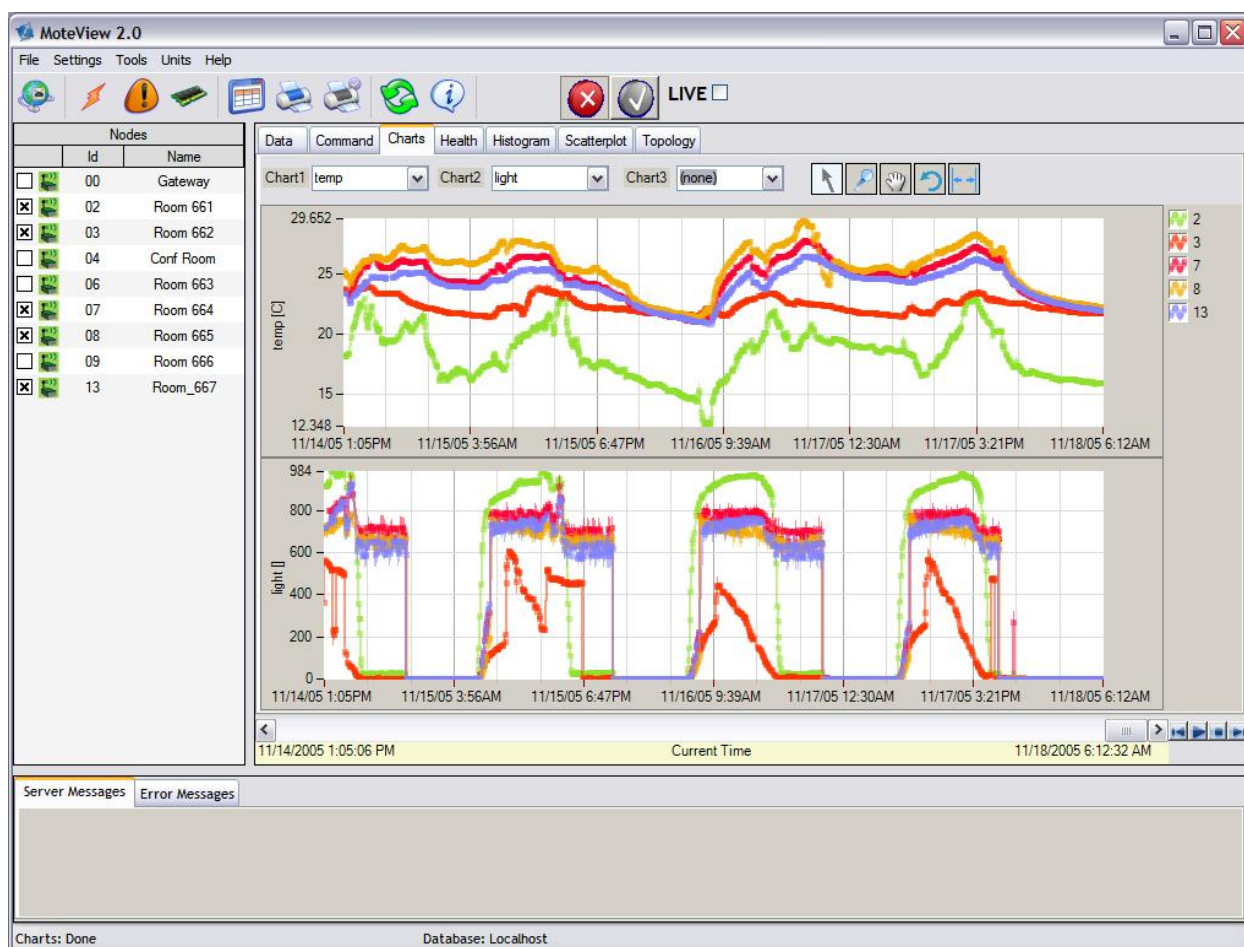






Figure 5-4. Screenshot of the demo database in the **Chart** tab

- The user can zoom into and pan through data, as instructed in Table 4-2, within each of the graphs independently of each other.

Table 5-1. How to zoom, pan, and reset in the Chart/Histogram/Scatterplot View.

Icon	Desired Action	Instructions
	To zoom in	Click on the "Zoom" icon. Left click and drag a region to zoom into. Release the mouse to complete the region selection.
	To pan through data	Click on the "Pan" icon. Left click and drag a point within the chart to the new location. Release the mouse button.

	To undo a zoom/pan	Click on the "Undo" icon once for each level of undo
	To zoom out fully	Click on the "View All" icon

- Right-clicking allows the user to select a fixed x-axis range (last hour, last day, last week, last month, last quarter, all data).

5.4 Health

The pre-compiled *XMesh* applications included with *MoteView* generate health packets at regular intervals. These health packets encapsulate the state of the wireless mesh over time. The **Health** tab displays the latest health packet readings received for each node in the network (see Figure 4-1 for an example from the demo database). The columns include node ID, health packet contents and server timestamp as described in Section 10.1. The sensor data is automatically converted into appropriate units (typically in %).

Left-clicking the column header allows you to sort by node ID, last result time, or any other health reading. Right-clicking the column header displays a pop-up menu with conversions relevant to the health field.

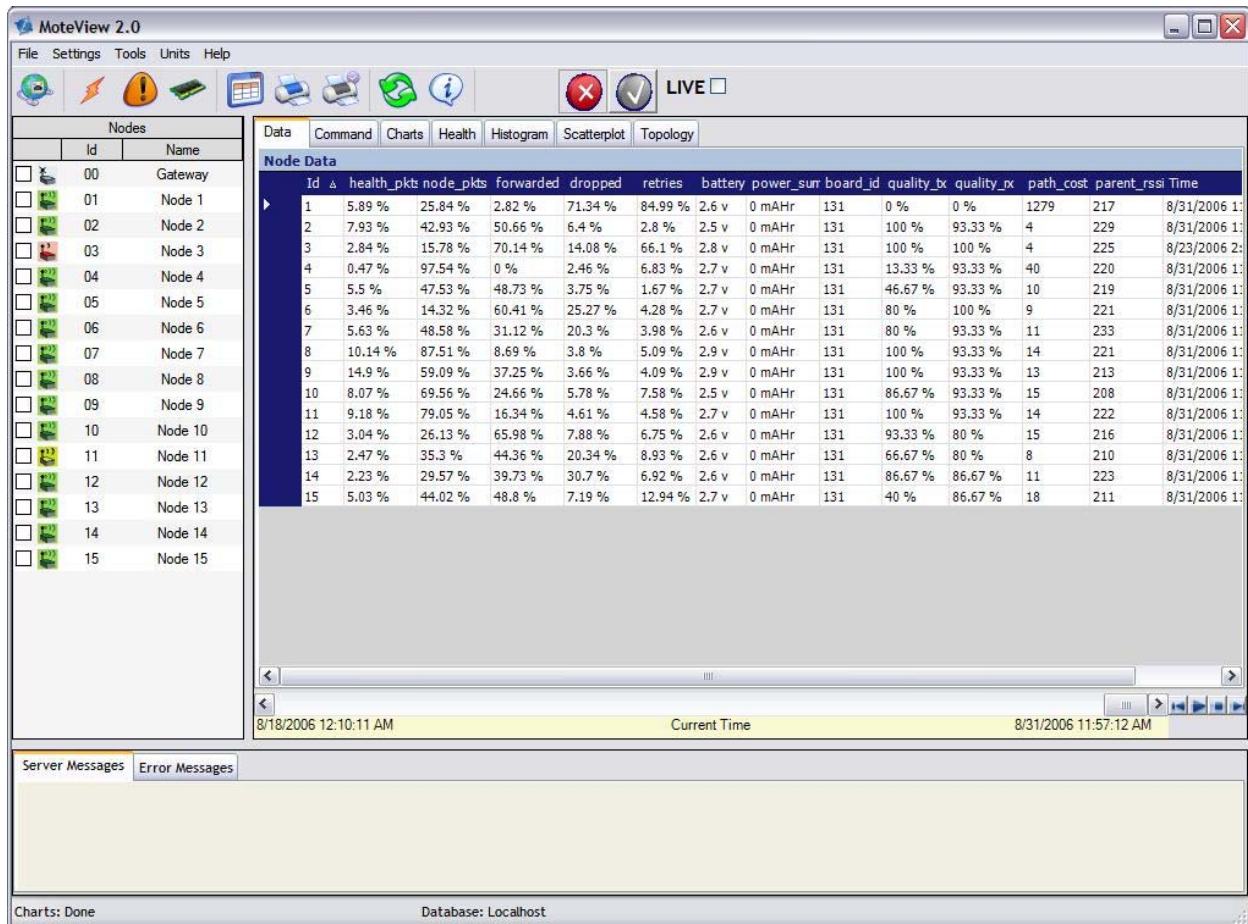


Figure 5-5. Screenshot of the demo database in Health view

5.5 Histogram

The **Histogram** tab shows a bar chart that graphically summarizes the statistical distribution of a single sensor data, allowing you see its center, spread and skewness etc. These features provide strong indications of the proper distributional model for the data.

- A single sensor can be selected from Sensor drop-down box for plotting,
- Up to 24 different Nodes can be selected for plotting by checking on the boxes next to the node list on the left side.. A different plot color will be used for each node; a legend is displayed on the right side of the window.
- The x-axis on the graph shows data in engineering units for the sensor values.
- The y-axis shows % for the instances for each of these sensor values.
- The user can zoom into and pan through data, as described in Table 4-2.
- Right-clicking allows the user to select a fixed x-axis range (last hour, last day, last week, last month, last quarter, all data).

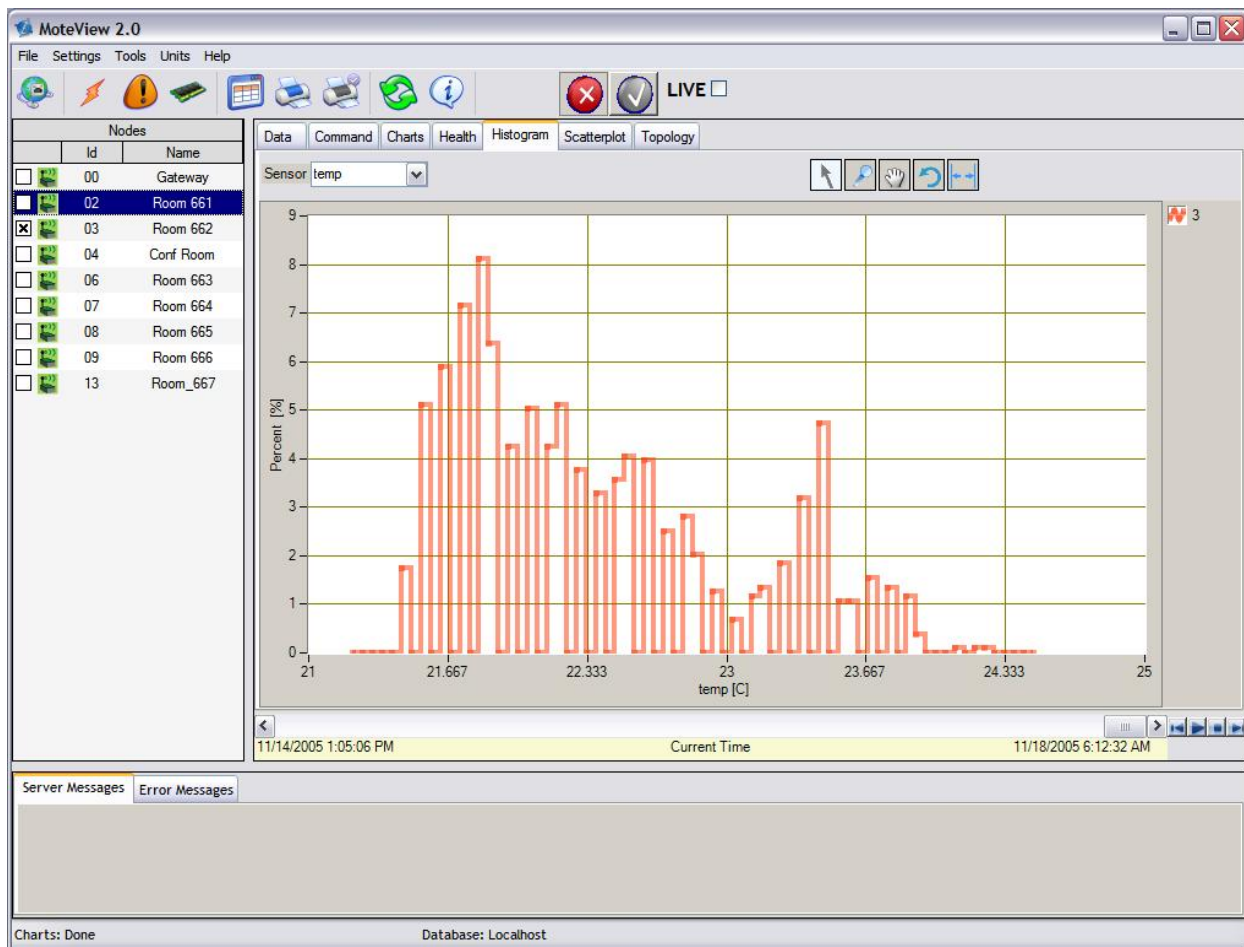


Figure 5-6. Screenshot of the demo database in Histogram view

5.6 Scatterplot

The **Scatterplot** tab provides the ability to plot two a sensor reading against each other for a set of nodes. The scatter plot would help obtain a visual comparison of the two sensor data fields and help to determine what kind of relationship there might be between them.

- The users can choose X and Y axis variables to be plotted from the drop-down boxes to apply to this view
- The user can zoom into and pan through data, as described in Table 4-2.
- Right-clicking allows the user to select a fixed x-axis range (last hour, last day, last week, last month, last quarter, all data).

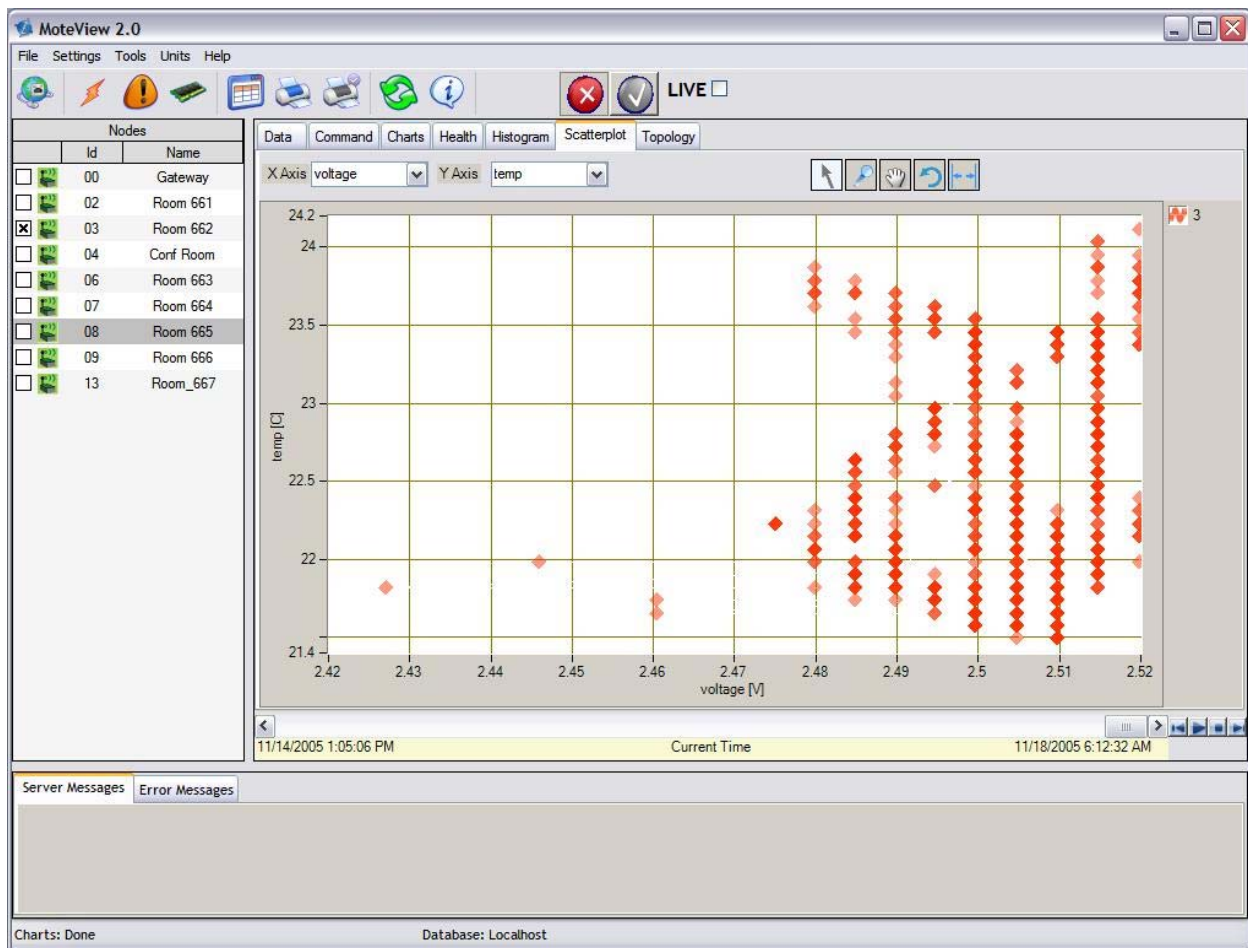


Figure 5-7. Screenshot of the demo database in Histogram view

5.7 Topology

The **Topology** tab shows a Mote network map with placement and parenting information. This allows the user to define and view a topology of their Mote deployment.

- New nodes will show up in the upper left hand corner.

- The user can drag nodes and place them at a new location on the map with a left click of the mouse. Node locations are stored in the database and are shared by all users of that database.

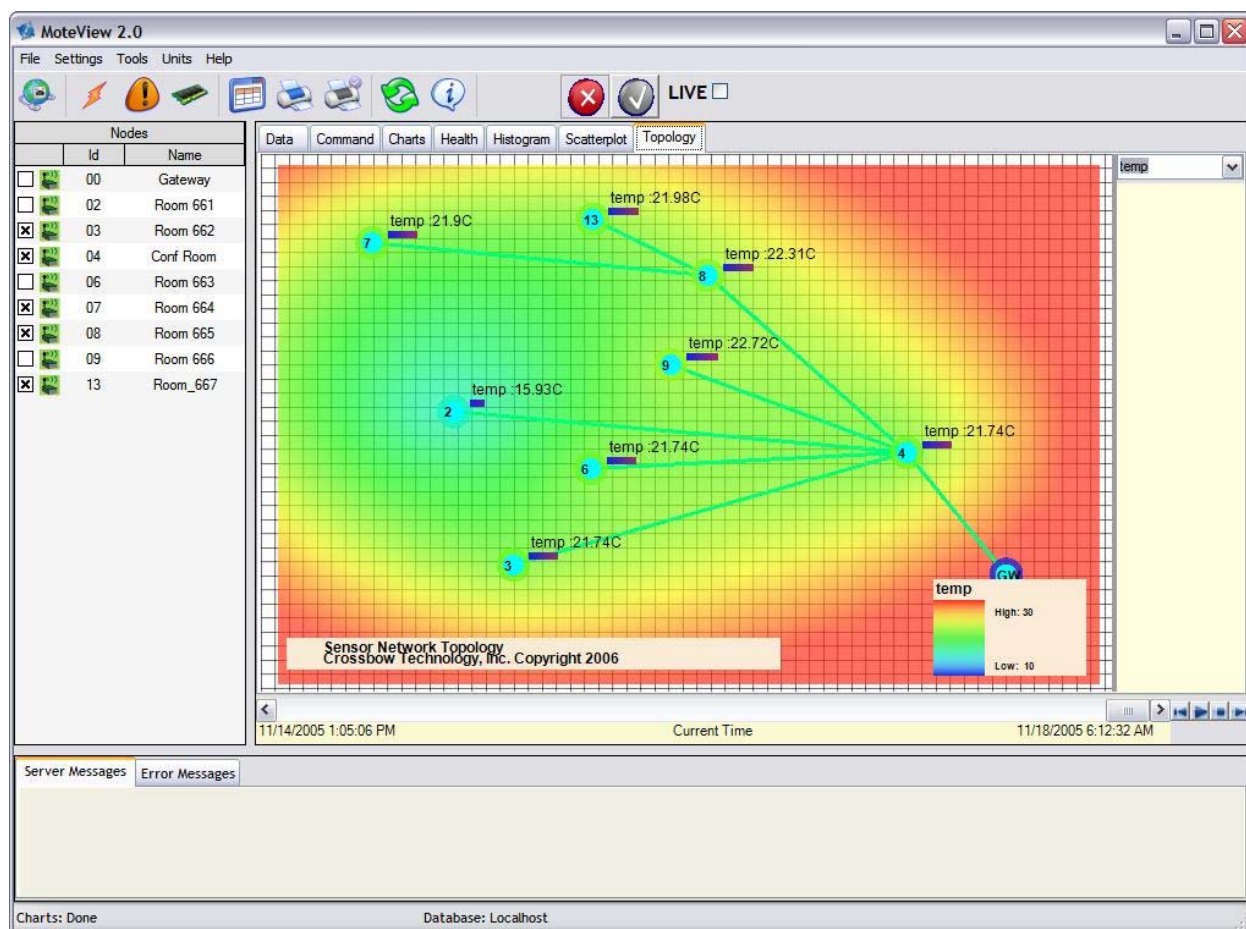


Figure 5-8. Screenshot of the demo database in Topology view

- A visualization pop-up menu will appear when the user right clicks the background bitmap.

Table 5-2. Items from the visualization pop-up menu that appears when right-clicking in the map area of the Topology view

Visualization Menu	Description
Add Node	Creates a new node at the current mouse location. A Mote properties dialog will appear to allow selection of node name and id number.
Arrange Nodes > Grid	Will automatically arrange all the nodes into a grid layout. This is useful when there are a large number of unplaced nodes.
Visualization Properties	Allows the user to visualize temperature gradients or other Mote properties using a specific color pattern.
MSP Properties	Sets the orientation of quadrant 1 relative to image in the topology view and the draw radius of a PIR event (Applicable for <i>MSP410</i> users)
Load Bitmap	Allows users to select a bitmap from their file system to be the background of the map. Supported background image formats include: .bmp, .gif, .ico, and .jpg. The image will be automatically scaled to fit the available screen space.
Use Default Bitmap	Use the standard grid as the background of the topology view.

The Visualization Properties lets users customize the following:

Sensor Color Gradient – Allows users to specify the minimum and maximum sensor values and associate specific color with it.

1. Right-click on the Topology bitmap and select Visualization Properties
2. Click on the Sensor tab of the Visualization Properties
3. Select a Sensor type from the drop-down list
4. Specify desired Maximum and Minimum sensor values
5. Click on the square colored box next to Maximum Color or Minimum Color
6. This will bring up another dialog box
7. Move the square marker on to choose the desired color and click OK.

The Sensor info also lets users display the sensor readings next to the node on the topology map and specify the font for the text.

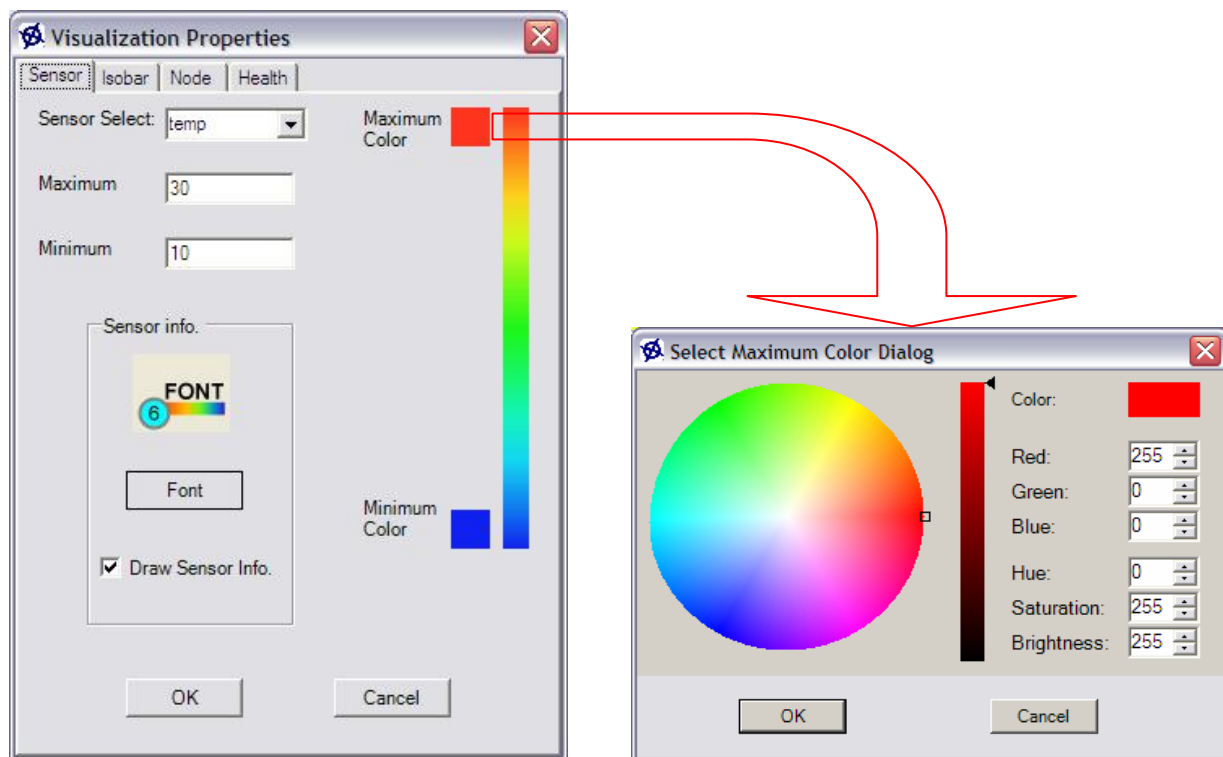


Figure 5-9. Screenshot for Sensor Gradient Visualization Dialog

Mote Visualization – Users can choose from 3 different display styles for nodes.

None – No nodes are displayed

BlackDot – A black dot is displayed to represent the node

MoteGlow – A colored circle is displayed along with the node ID. The circle's color is based on the current sensor value and the minimum and maximum sensor values specified in the visualization properties.

Users can also check options for Draw Links between nodes and specifying whether Gateway has a sensor on it for gradient visualization.

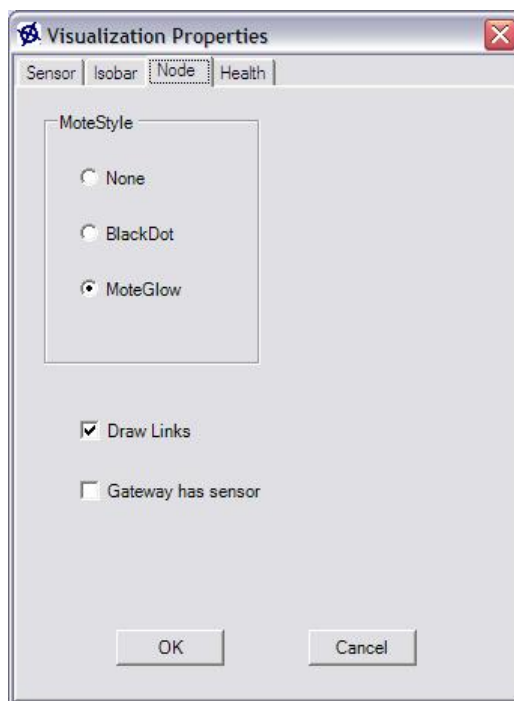


Figure 5-10. Screenshot for Mote Visualization Dialog

To view a color gradient superimposed on the topology view, select the **Draw Gradient** radio button in the **Isobar Visualization** box.

Users can also specify the **Mote Radius** (as a percentage of bitmap image size; 0=0%, 10=100%) for spread of gradient around the node; check options for displaying the ScaleBar Legend and drawing the Logo.

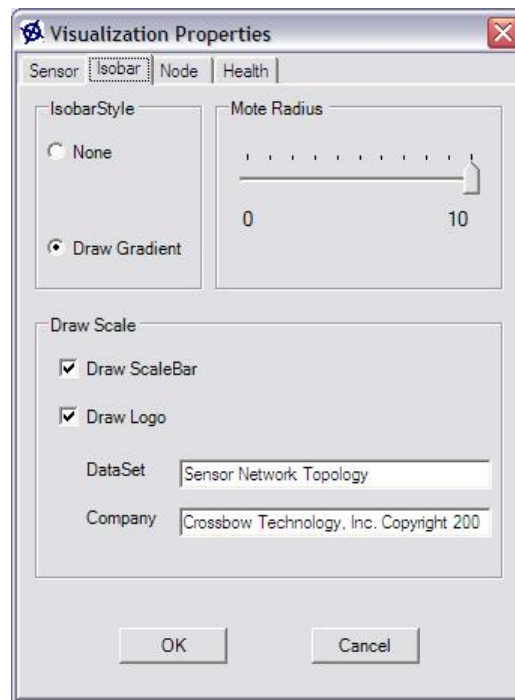


Figure 5-11. Screenshot for Isobar Visualization Dialog

Health Visualization - Users can specify the time duration after which the link between the nodes goes grey. If a packet is not received from any after the specified minutes, the link would turn grey.

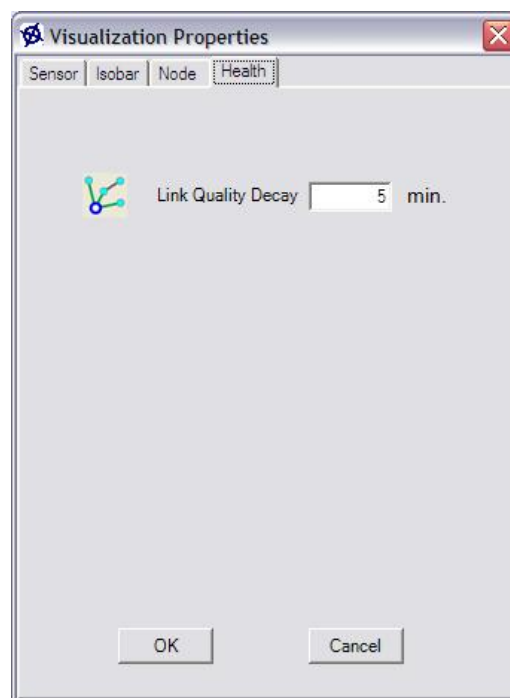


Figure 5-12. Screenshot for Link Quality Visualization Dialog

5.8 Live/Historical/Playback Mode

In order to have *MoteView* display data from an active sensor network, the users must check on the **Live** checkbox. In Live mode, *MoteView* refreshes the node list, charts, and topology views as packets are received at the base station.

If **Live** is unchecked, the **time-bar** at the bottom of the visualization tabs will be enabled. This allows users to scroll back and forth and view data collected at different times. The **time-bar** interface is supported on the **data** and **topology** tabs.

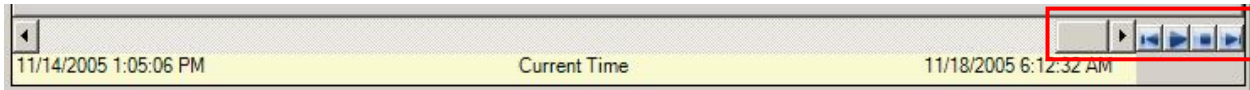


Figure 5-13. Screenshot showing a portion of the MoteView window to highlight the Timebar

On the right-hand side of the **time-bar** are a collection of **playback** controls. Scroll the time-bar to a specific time and press the play button to view an animation of the data collected over time. Right-clicking on these controls will open a settings dialog that allows the user to set the playback interval.

6 Toolbar/Menus

The *MoteView* menu area contains the **File**, **Settings**, **Tools**, **Units**, **Help** menus and a toolbar providing quick access to some of the more common features of the *MoteView* application. From left to right, the toolbar provides buttons for **Connect to WSN** wizard, **Program Mote** (*MoteConfig*), **Alerts Manager**, **MDA configuration**, **Export to Spreadsheet**, **Print**, **Print Preview**, **Refresh View**, **MoteView Help**, **Stop XServe**, **Start XServe** and checkbox for displaying **Live** (vs. historical) sensor network data.



Figure 6-1. Screenshot showing a portion of the MoteView window to highlight the icon toolbar menu

This chapter will describe the features in each of these menu items.

6.1 File > Connect to WSN

This will bring up the Connect to WSN wizard, which enables the user to specify the Operation Mode, Acquisition Type, Gateway, Database, and Sensor Application. This wizard provides 4 different tabs.

6.1.1 Mode

This allows users to select Operation Mode and Acquisition Type.

Operation Mode can be

- Live Data – View data from an active network
- Historical Data – View logged data

Acquisition Type can be

- Local – The gateway connected directly to the local PC via a MIB board and the database runs on local PC (localhost)
- Remote – The gateway will be a Stargate or remote server and the database resides on this remote gateway
- Custom – The gateway will be a remote server and the database can be either mirrored locally (localhost) or can reside on another remote server.

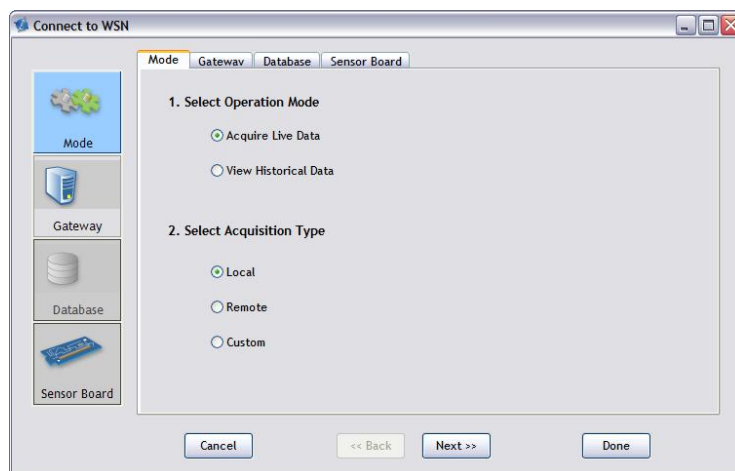


Figure 6-2. Mode Tab of the Connection Wizard

6.1.2 Gateway

This allows users to specify the Gateway and Port settings. The available options are:

- MIB510 – Serial gateway
- MIB520 – USB gateway
- MIB600 – Ethernet gateway
- XServe – Connection to a remote XServe

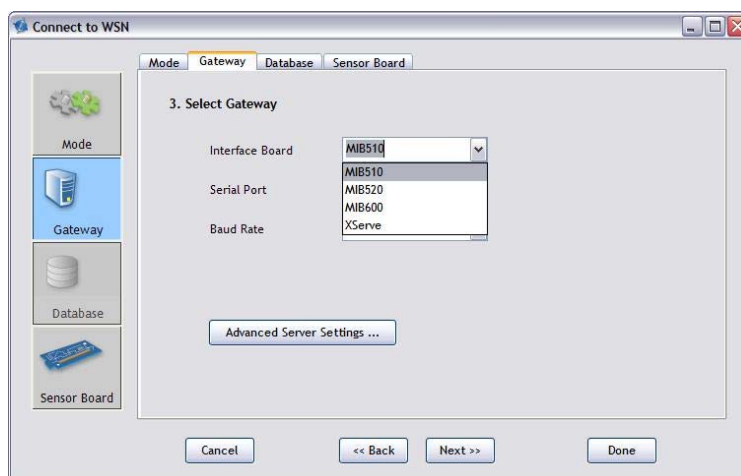


Figure 6-3. Gateway Tab of the Connection Wizard

Pressing the **Advanced Server Settings...** button opens the Advanced Logging Dialog.

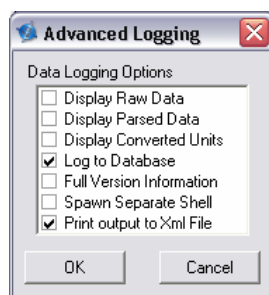
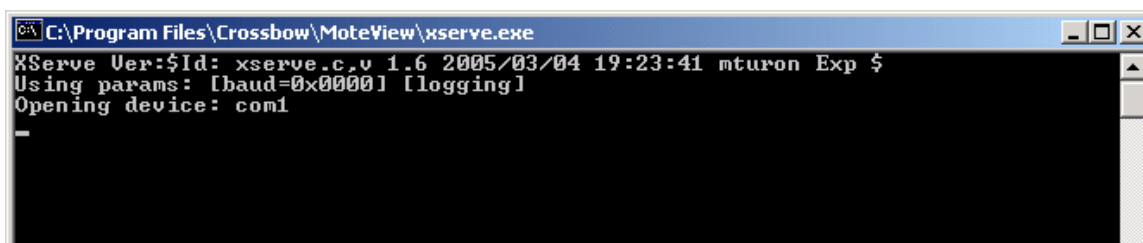


Figure 6-4. Advanced Logging Dialog Window

Table 6-1. Description of advanced logging options and the equivalent XServe flag

Advanced Logging Option	Description	XServe flag
Display Raw Data	The raw packet bytes coming into the PC.	-r
Display Parsed Data	The raw ADC values displayed as parsed fields.	-p
Display Converted Data	The incoming data converted to engineering units.	-c
Log to Database	Will store raw ADC values to PostgreSQL database.	-l
Full Version Information	Display exact version for all board modules	-v
Spawn Separate Shell	Launches XServe in command line shell	n/a

The **Start XServe** and **Stop XServe** buttons are also provided in the icon toolbar for easy access. The **Start XServe** button will be enabled if *XServe* is not already running. Clicking on **Start XServe** will initiate *XServe* as a separate process with the given options. The output of *XServe* will be streamed to the **Server Messages** pane. If **Spawn Separate Shell** has been selected, the *Xserve* output will be displayed in the *Command* window, as shown in Figure 6-5.

Figure 6-5. Portion of a *Command Prompt* window that appears once the data logging has started

If **Log to Database** is selected, *Xserve* will output the SQL statements used to insert the contents of each data packet as they arrive. Following is an example of the MTS400 and MDA300 SQL statements.

XMTS400 / XMTS420 Logging Output

```
INSERT into mts420_results
(result_time,nodeid,parent,voltage,humid,humtemp,inte
rsemacal1,intersemacal2,intersemacal3,intersemacal4,prtemp,press) values
(now(),
0,126,377,1391,6517,45446,46808,38941,45989,26835,18130)
```

XMDA300 Logging Output

```
INSERT into mda300_results
(result_time,nodeid,parent,voltage,humid,humtemp,adc0
,adc1,adc2,digi0,digi1,digi2) values
(now(),0,126,377,960,6678,2500,2243,2505,0,0,0)
```

6.1.3 Database

This allows users to configure database settings. Saved configurations are stored in the tree view to the left and the current database gets highlighted. Clicking on a database name loads those settings into the fields on the right. Users can Add, Edit or Remove these entries.

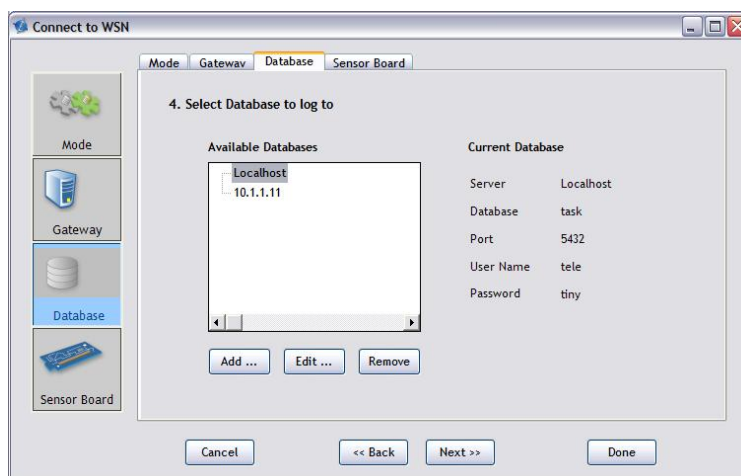


Figure 6-6. Database Tab of the Connection Wizard

Clicking on Add... or Edit... buttons bring up the dialog box shown in Figure 6-4.

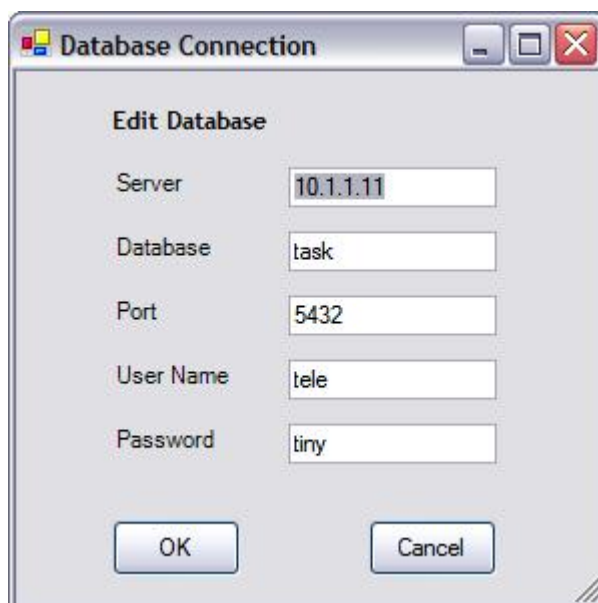


Figure 6-7. The Database Server Configuration dialog box

Specify the following fields:

- **Server:** Can be a hostname or IP address. To specify your workstation as a local server, enter `localhost`.
- **Database:** The name of the database, defaults to `task`.
- **Port:** TCP/IP port of *PostgreSQL* database server. Defaults to `5432` which is the standard *PostgreSQL* port.
- **User:** username for database server - `tele` is the default.
- **Password:** password for database server - `tiny` is the default.

After inputting these five fields above, click the **OK** button to attempt to connect to the specified database.

6.1.4 Sensor Board

This allows users to specify the sensor application type that is programmed into the Motes or the table name in the database to connect to.

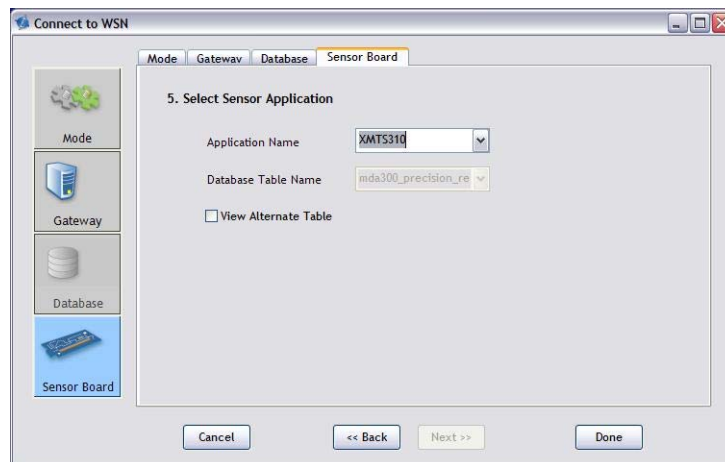


Figure 6-8. Sensor Board Tab of the Connection Wizard

Users can specify the Sensor Application from the **Application Name** dropdown box or View Alternate Table by checking the checkbox and selecting from **Database Table Name** drop-down.

6.2 File > Import

MoteView can import data from a previously exported SQL data file. This will modify the contents of your database

6.3 File > Export

You can export data into 5 different formats

- *Export Database to SQL* – exports the entire task database to an SQL file
- *Export Table to SQL* – exports the contents of the current table in the task database to an SQL file
- *Export Table to Spreadsheet* – exports the contents of the current table in the task database to a spreadsheet file (delimited text file). The file delimiter (comma, tab or space), Nodes (all nodes or only a subset of nodes) and Time range (all data or time range) can be specified in the dialog window as shown in Figure 6-9.

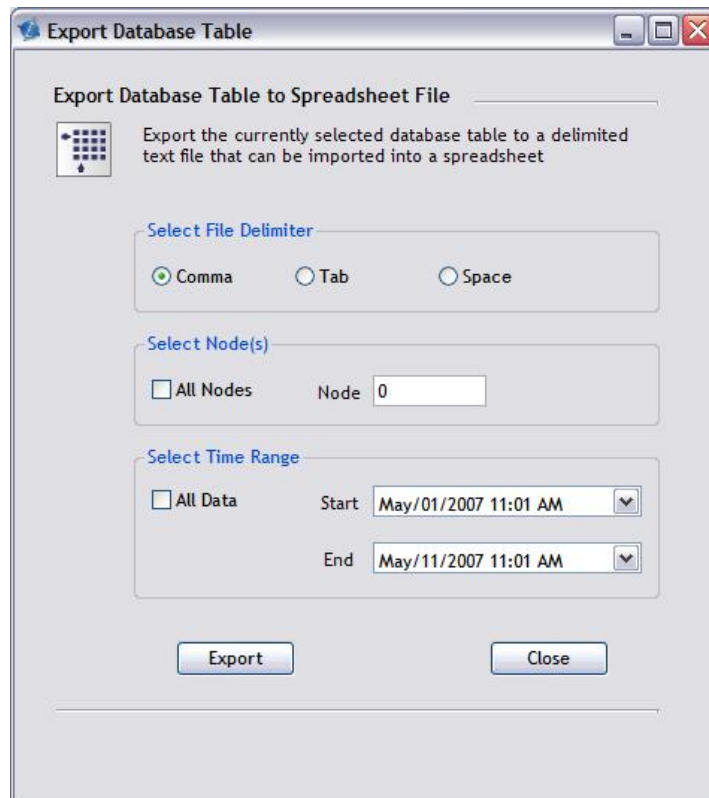


Figure 6-9. Export Database Table Dialog Window

IMPORTANT

Exporting large sets of data may cause slow-down of your PC and freeze MoteView. It is recommended that you exports the data in chunks by specifying the node ID or Time range.

- *Export Table to XML* – exports the contents of the current table in the task database to an XML file.
- *Export Active Tab to JPEG* – exports an image of the current tab view to a JPEG file

6.4 File > Print Preview

The Print Preview screen will display the expected output for the Print command for the active visualization tab. The magnifying glass icon allows you to zoom in on the preview. The printer icon will immediately print out the page to the default printer.

6.5 File > Print

This menu item prints the active visualization tab.

6.6 File > Exit

This menu option will exit the application and close *MoteView*. *MoteView* always attempts to save the current configuration when closed and restore that configuration the next time the application is started.

6.7 Settings > General

- *Server Heartbeat* – The user can select how many heartbeat messages are displayed in the Server Messages window. These messages are sent by the XMeshBase firmware at 5 second intervals.
- *Toolbar Style* – The user can choose from Simple, Modern and Cartoon style for toolbar display.
- *Playback settings* – The user can set the speed at which historical data is displayed when using the playback controls on the right-hand side of the time-bar.

6.8 Settings > Node

- *Health Status* – Opens the node health settings window described in Section 4.
- *Properties* – Opens a node properties window for a selected node (see Section 4.3)

It allows the user to assign a **Name** and **Group** of the Mote.

The **Color** tab allows the user to select the charting color for the Mote.

6.8.1 Spreadsheet Delimiter

This feature specifies how the spreadsheet file (delimited text file) separates its data values. 3 delimiters are available (commas, tabs or spaces).

6.9 Tools > Refresh View

This will force a visual refresh of all visualization windows in *MoteView*. After clicking on this menu the **Nodes** list, the **Chart**, and **Topology** views will all update to display the latest information from the sensor network.

6.10 Tools > Add Node

This will bring up the Mote Properties dialog which allows the creation of a new node with a user-specified Node ID and name (see Section 4.1).

6.11 Tools > Program Mote

This opens a separate application called *MoteConfig* which allows users to download firmware to a Mote over a gateway and over the air. Details are available in the *MoteConfig User's Manual*.

6.12 Tools > Alerts

The *MoteView* alert manager allows users to define alert conditions based on any sensor data of any sensor node. An alert is a user programmable event that gets triggered when sensor data exceeds pre-defined threshold. An alert is composed of several pieces:

1. **Alert ID** – A sequential number to identify the alert
2. **Node Name** – the node whose conditions can trigger an alert
3. **Sensor Name** – the sensor whose readings can trigger an alert
4. **Alert Condition** – the comparison operation to decide when an alert should be triggered
 - Equals
 - Exceeds
 - Below
 - Exceeds or Equals
 - Below or Equals
5. **Alert Threshold** – the value to compare the alert condition against
6. **Unit** – the unit for the alert threshold
7. **Alert Action** – the operation to perform in response to a triggered alert
 - Display a message dialog and console message
 - Send an email (allows text paging a mobile phone)
8. **Alert Interval** – specify the interval for the action to be taken.
9. **Alert Duration** – specify the time interval of the alert condition before the alert should be triggered.

To set an Alert, follow these steps below:

1. Chose from **Tools > Alerts > Alert Manager...** and click the button “**Add New Alert Item**”. This operation will add a blank alert item to the list with Alert ID 1 as shown in Figure 5-8.

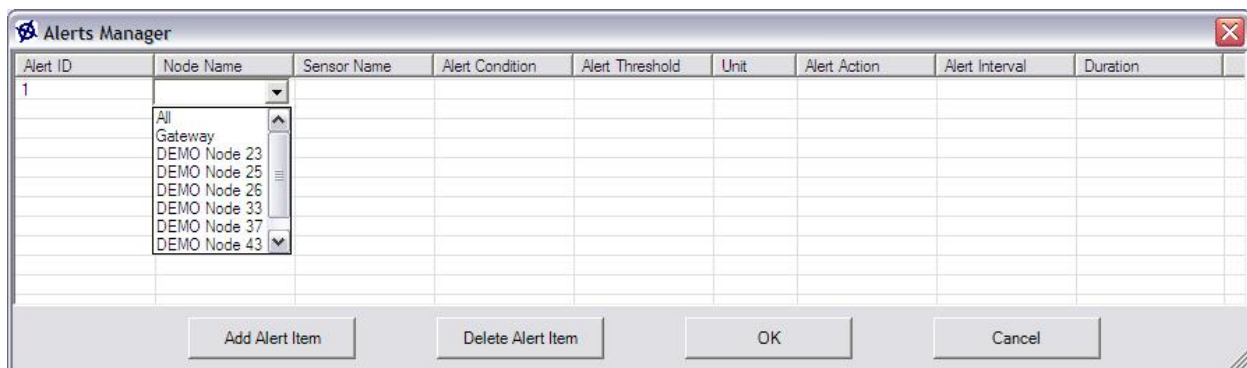


Figure 6-10. Screenshot of the Alert Manager

2. From the **Node Name** drop-down list, select a node that is currently a part of your deployment.

3. From the **Sensor Name** drop-down list, select a sensor that you want to use to trigger the Alert. This automatically updates the Unit column associated with that particular sensor type.
4. From **Alert Condition** drop-down list, specify an appropriate condition to trigger the Alert, viz. "<", ">", "<=", ">=", "=".
5. For **Alert Threshold** field, input numerical value in the text box.
6. From **Alert Action** drop-down list, chose an action, "Send Alert Email" or "Pop-up Alert Form".
7. From the **Alert Interval** drop-down list, specify the interval for the Alert to be sent.
8. From **Duration** drop-down list, specify the time duration for which the Alert Condition needs to be met before the Alert Action can be triggered.

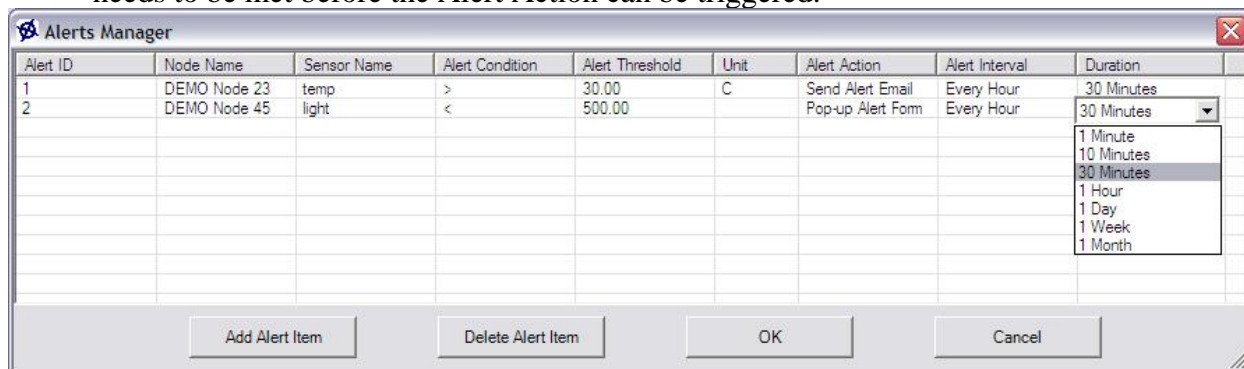


Figure 6-11. Screenshot of the Alert Manager after adding two Alerts

To delete an alert, select and highlight the item that you wish to delete then click the "Delete Alert Item" button, the selected alert item will be deleted.

For the Alert manager to send Email, the users should first configure the mail settings. This can be accomplished as follows:

1. Click from **Tools > Alerts > Alert Mail Configure...** and that will bring up the dialog box shown in Figure 5-10
2. Enter the **SMTP Server** name of your mail server
3. Specify the **User Name** and **Password** of your mail account
4. In **Send To** field, specify the Email address where the alert needs to be sent
5. Enter any optional **Message** to be sent in the body of the Email
6. Click on **OK**.

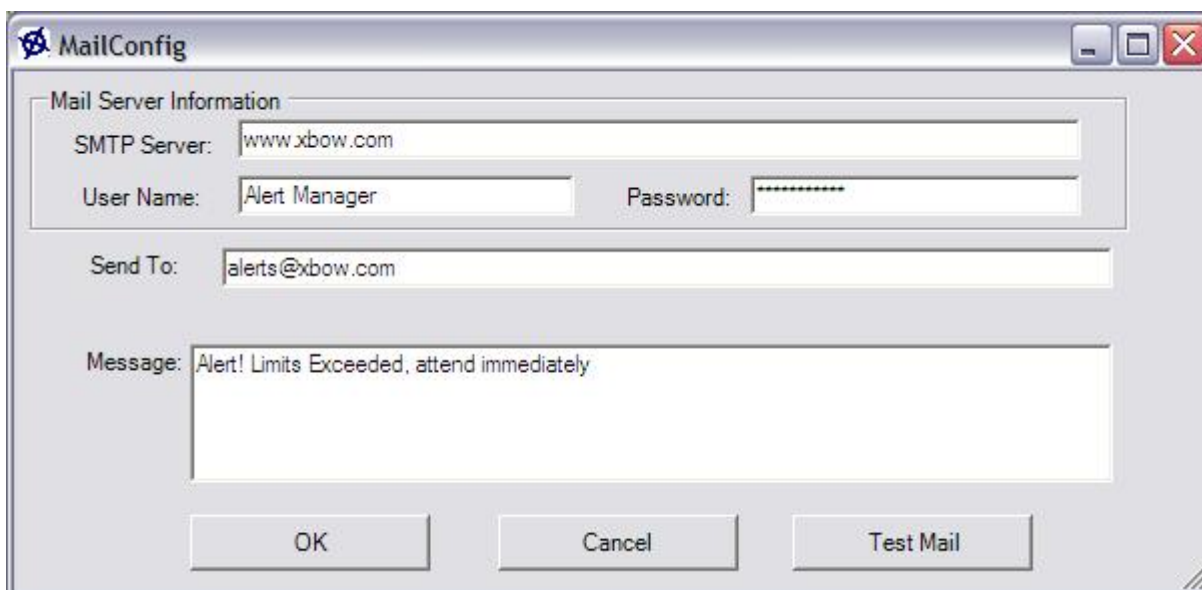


Figure 6-12. Screenshot of the Alert Mail Configuration

6.13 Tools > MDA Support

MoteView allows users to define their own calibration parameters to convert from raw voltage into engineering units for external sensors when connected via MDA series data acquisition boards. Users can specify the calibration coefficients for each of the available channels.

MDA Support interface allows the users to define custom external sensors, conversions, and associate them with channels on a data acquisition board. It is composed of several pieces:

- MDA Board Type – MDA100, MDA300, MDA320 or MDA325, MDA500
- Sensor Type – Configuration name associated with an ADC channel
- Configuration Name – assigned to a particular sensor
- Units – Engineering units associated with a particular configuration/conversion
- Conversion type – Third order polynomial, look up table or thermistor

To create a new external sensor configuration, follow these steps:

1. MDA Configuration dialog window can be started by clicking on **Tools > MDA Support**. This will bring up a window shown in Figure 5-11.

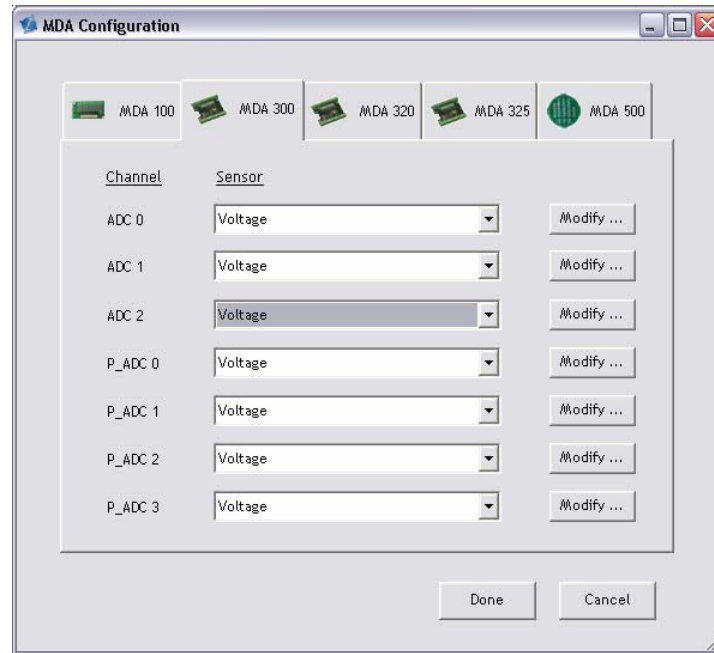


Figure 6-13. Screenshot of the MDA Configuration Support

2. Select the specific MDA board type that you are currently using to connect external sensors.
3. Next to the Channel of interest, from the **Sensor** dropdown box see if any of the choices is relevant to you. If not, click on **Modify...** This will bring up a dialog window for Sensor Configuration as shown in Figure 5-12.

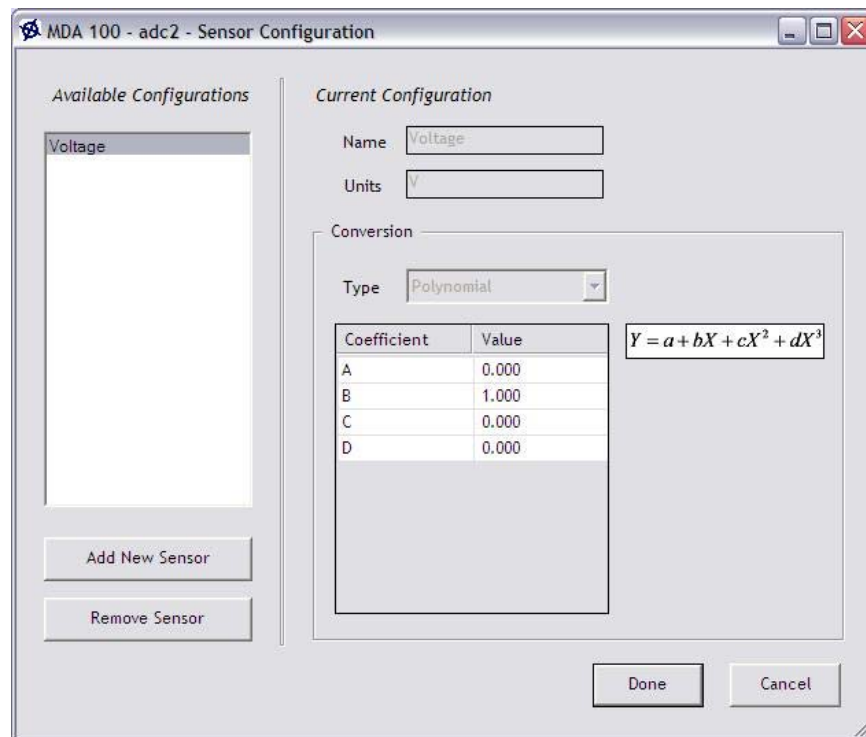


Figure 6-14. MDA Sensor Configuration

4. Click on **Add New Sensor**.
5. Edit the **Name** and **Units** under Current Configuration group box.
6. Under Conversion group box from the **Type** dropdown, select the calibration type (i.e. **Polynomial**, **Look Up Table** or **Thermistor**).
 - a. If you used **Polynomial** as the conversion type, then specify the Coefficient values (a, b, c and d) to convert from raw voltage (X) into engineering units (Y) as shown in Figure 5-13.

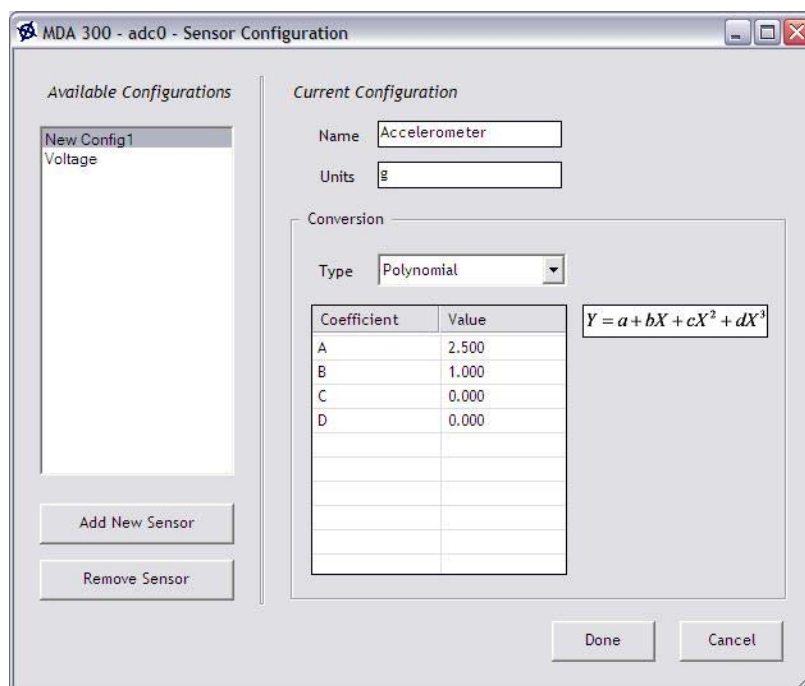


Figure 6-15. MDA300 Sensor Configuration using the Polynomial Conversion

- b. If you used **Look Up Table** as the conversion type, then input the **Voltage** readings and corresponding engineering values in the table as shown in Figure 5-14. Users can add or delete a row using the **Add Row** or **Delete Row** buttons.

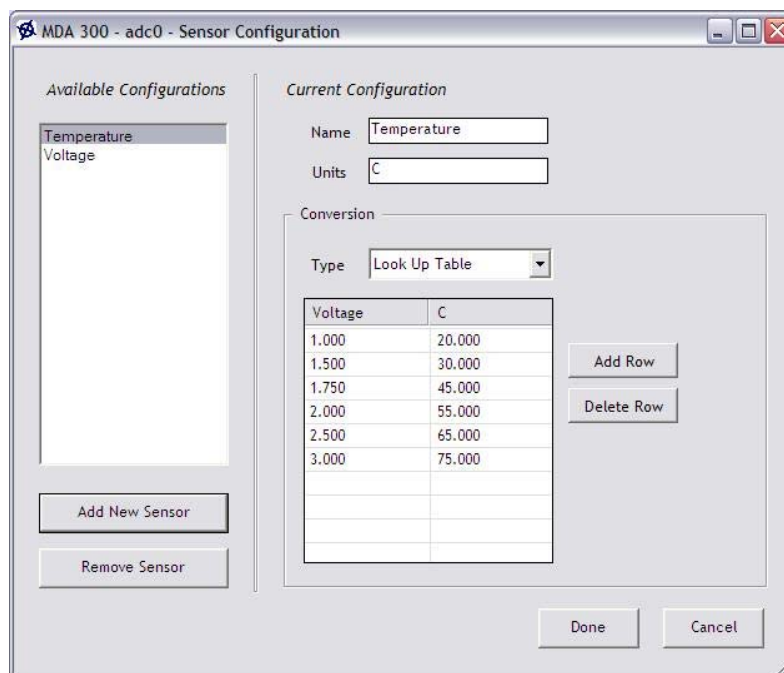


Figure 6-16. MDA300 Sensor Configuration using the Look Up Table Conversion

- a. If you used **Thermistor** as the conversion type, input the excitation voltage, external fixed resistance and the Steinhart-Hart coefficients as shown in Figure 5-15. Note that the converted temperature is only available in degrees C.

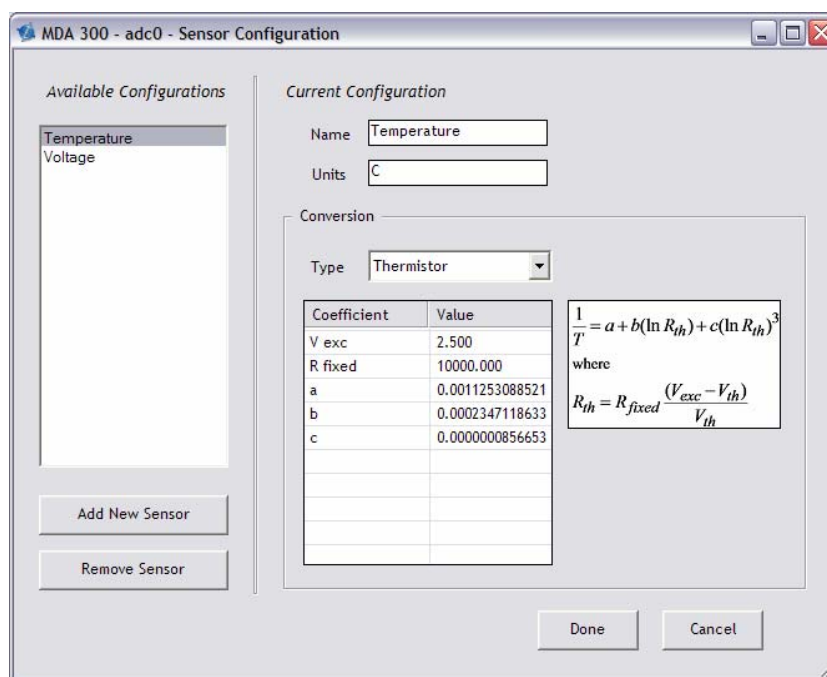


Figure 6-17. MDA300 Sensor Configuration using the Thermistor Conversion

7. Click on **Done** for the new Sensor Configuration to take effect.
8. This saved Sensor Configuration will now be available in **Sensor** dropdown box of Figure 5-11 and can then also be used for other ADC channels.
9. Users can remove a particular sensor configuration by highlighting it and then clicking on **Remove Sensor**.

6.14 Tools > Reset Database

This executes a script that will delete all data from the local database and reinitialize it. This feature is not supported for remote databases.

6.15 Units

The units menu allows the user to select their preferred engineering units for a given class of sensor. All sensors of that class will convert to the units specified by the user. A user may select to view raw data as direct ADC readings instead of a particular unit. The units for a specific sensor can be changed by right-clicking on the **column header** in the **data** tab.

6.16 Help > User's Manual

This menu item opens the *MoteView User's Manual*.

6.17 Help > Support

This menu item opens a browser window to the Crossbow support webpage if your PC is connected to the internet.

6.18 Help > About

This displays a splash screen containing *MoteView* version information.

7 Server/Error/Status Messages

The **Server Messages** pane is the bottom section of the *MoteView* window and displays server side messages, error messages, and general status messages as *MoteView* is running.



Figure 7-1. Screenshot showing a portion of the MoteView window to highlight the Server, Error and Status Messages

The **Server Messages** tab displays the *XServe* messages, raw/converted packets and database insert statements.

The **Error Messages** tab displays exceptions/errors reported by *MoteView* software.

The **Status bar** at the bottom displays chart query status, current database and gateway settings in a plain view.

- On the left hand side, it displays the Charts query (Loading... or Done).
- At the center, it displays the Database and table name.
- On the right hand side, it displays the gateway type and port settings.

8 Server Administration

XServe is a powerful packet parsing and data logging server preinstalled by *MoteView*. A user may run *XServe* from a *Cygwin* shell (included in the MoteWorks CD).

❏ **NOTE:** To use the command line interface for *XServe* you **must** have the *Cygwin* shell installed on your PC. Instructions for installing *Cygwin* from *MoteWorks* InstallShield Wizard are detailed in the *MoteWorks Getting Started Guide*.

For proper operation, the `cygwin1.dll` library in the *MoteView* installation directory must exactly match the `cygwin1.dll` library in the user's `cygwin/bin/` directory.

MoteView starts *XServe* when a logging session is started as described in sections 3.2 and 5.1.1. To understand how *XServe* can be used from a *Cygwin* shell, type `xserve -?` to display the command line options.

```
Usage: xserve <-?|r|a|p|c|xr|xp|xc|dbxmlr|xmlp|xmlc|v|alert|m>
        <-l=tablename>
        <-dbserver=servername> <-dbport=portnum>
        <-dbname=database name> <-dbuser=username> <-dbpasswd=password>
        <-h=path,hostname,portnum,config_file>
        <-m=com,baud,protocol,slaveaddress,defaultregistervaluesas>
        <-xmlfile=filename> <-xmlport=portnum>
        [<-sf=hostname:port> | <-fsf=hostname:port> | <-device=dev>]
        <-port=num> <-baud=num> <-platform=plt>
        <-debug=level>
        <-configfiles=filename:filename:>
        <-loadparsers=filename:filename:...>
        <-loaddatasinks=filename:filename:...>
        <-heartbeat=<num missed>

-?      = display help [help]
-r      = raw display of tos packets [raw]
-a      = ascii display of tos packets [ascii]
-p      = parsed display of tos packets [parsed]
-c      = converted display of tos packets [conveted]
-xr     = raw tos packets xported to file [export raw]
-xp     = parsed tos packets exported to file [export parsed]
-xc     = converted tos packets exported to file [export converted]
-db     = parsed tos packets exported to db [database parsed]
-dbserver = database server name (default=localhost)
-dbport  = database server port number (default=5432)
-dbname  = database name (default=MoteView db)
-dbuser  = database user (default=MoteView user)
-dbpasswd = database user password (default=MoteView user password)
-l      = parsed tos packets exported to db
        (deprecated) [database parsed]
-xmlr   = raw tos packets exported to xml [xml raw]
-xmlp   = parsed tos packets exported to xml [xml parsed]
-xmlc   = converted tos packets exported to xml [xml converted]
-xmlfile = file name to store exported xml (default=screen)
-xmlport = port number to start the xml server
-v      = show version of all modules
```

```
-h          = display data through web server
-m          = export data using modbus
-port       = set server port <default = 9001>
-sf         = connect to unframed serial forwarder
-fsf        = connect to framed serial forwarder
-device     = connect to serial device <default = /dev/ttyS0>
-baud       = set serial baud rate <default = 57600>
-platform   = set platform. <default = mica2>
              values=mica2dot|mica2|mica|telos|micaz
-debug      = set debug level. <default = DBG_WARNING>
-alert      = alert when data values are above/below specified ranges
-daemon     = run in daemon mode
-nomonitor  = run without a system monitor
-configfiles = load xml configuration files.
-loadparsers = load only the listed parsers files from the
              shared_lib. (default=all files are loaded)
-loaddatasinks = load only the listed datasinks files from the
              shared_lib. (default=all files are loaded)
-heartbeat  = turn on the heartbeat monitor and reset after <num
              missed>
-convZto2   = convert incoming network packets from micaZ headers to
              mica2 headers and vice versa
-conv2toZ   = convert incoming network packets from mica2 headers
              to micaZ headers and vice versa
-joinfile   = node mapping file for unique ids joining the network
-noorganic   = disable organic growth for joining nodes
```

Refer to the *XServe* User's Manual for more details.

9 Database Administration

9.1 PostgreSQL

PostgreSQL is an advanced relational database system included in the MoteView installer. The database tables that *MoteView* accesses can be manipulated directly by advanced users. To access the *PostgreSQL* database, from a *Cygwin* shell type

```
psql -h localhost -U tele task
```

Below is an example of what you should get as a response to that command:

```
$ psql -h localhost -U tele task
Welcome to psql 7.4.5, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

task=#
```

You can also use the psql command line interface in the *Windows Command* window from the C:\Program Files\PostgreSQL\8.0.0-rc1\bin folder.

9.2 SQL

SQL is the generic command language used to manipulate databases such as *PostgreSQL*. SQL commands can be typed in directly from the *PostgreSQL* command shell. A list of useful commands follows:

9.2.1 Display all readings

Type

```
select * from <tablename>;
```

The **select** statement will display results out **from** the given <tablename>. The * character is a wildcard meaning that all columns should be displayed.

```
task=# select * from surge_results;
      result_time      | epoch | nodeid | parent | light | temp | voltage |
mag_x | mag_y | accel_x | accel_y
-----+-----+-----+-----+-----+-----+-----
2004-07-12 12: 47: 54.29425 | 757 | 4 | 0 | 182 | 135 | 229 |
255 | 255 | 200 | 200
2004-07-12 12: 48: 02.26325 | 758 | 4 | 0 | 181 | 135 | 229 |
252 | 252 | 200 | 200
```

9.2.2 Display subset of readings

Type

```
select field1, field2,... from <tablename> where fieldn condition value;
```

The **select** statement will display results **from** the given <tablename>. The * character is a wildcard meaning that all columns should be displayed.

```
task=# select result_time, temp, light from surge_results where nodeid=1;
      result_time      | temp | light
-----+-----+-----
2004-07-12 12: 48: 07.31025 | 136 | 234
2004-07-12 12: 48: 15.29325 | 136 | 232
```

9.2.3 Rename a table

Type

```
ALTER TABLE <tablename> RENAME TO <newname>;
```

9.2.4 Delete all readings from table

Type

```
DELETE FROM <tablename>;
```

9.2.5 Deleting specific readings from table

To delete all results before the specified date, type

```
DELETE FROM <tablename> WHERE result_time < '2004-11-20';
```

To delete all results with ADC voltage reading greater than 400, type

```
DELETE FROM <tablename> WHERE voltage > 400;
```

To delete all results from node number 3, type

```
DELETE FROM <tablename> WHERE nodeid = 3;
```

9.2.6 Delete table entirely

Type

```
DROP TABLE <tablename>;
```

9.3 Database Tools

PostgreSQL includes tools for offline manipulation of data besides the *psql* shell. The more useful of these are described here. These windows command prompt version of these tools are installed to *C:\Program Files\PostgreSQL\8.0.0-rc1\bin* by default. The *psql* tool is available from the *Cygwin* command prompt as well.

9.3.1 PostgreSQL Export

To output the entire task database to a file, (e.g., *my_database.out*)

```
pg_dump -h localhost -U tele -f my_database.out task
```

To save the contents of *mts400_results* table to a file of SQL commands named *mts400data.out*:

```
pg_dump -h localhost -U tele -t mts400_results -f mts400data.out task
```

◀ **NOTE:** Exporting a database table or the entire database can also be performed by selecting the following from the MoteView menu-bar:

File>Export>Export Database to SQL

File>Export>Export Table to SQL

◀ **NOTE:** A database table can be exported to a spreadsheet file (tab, comma or space delimited), or XML file by selecting the following from the MoteView menu-bar:

File>Export>Export Table to Spreadsheet

File>Export>Export Table to XML

9.3.2 PostgreSQL Import

To load files from a PostgreSQL exported table, use the following command:

```
psql task < surge.out
```

◀ **NOTE:** Importing previously exported SQL files can also be performed by selecting the following from the MoteView menu-bar:

File>Import>Import from SQL

10 Health Packets

The pre-compiled *XMESH* applications included with *MoteView* generate health packets at regular intervals. These health packets encapsulate the state of the wireless mesh over time. The **Health** tab displays the latest health packet readings received for each node in the network (see Figure 10-1 for an example from the demo database).

In addition to the node health packet, each Mote also transmits a neighbor health packet. The latter is logged to the *postgreSQL* database but is not currently displayed by *MoteView*.

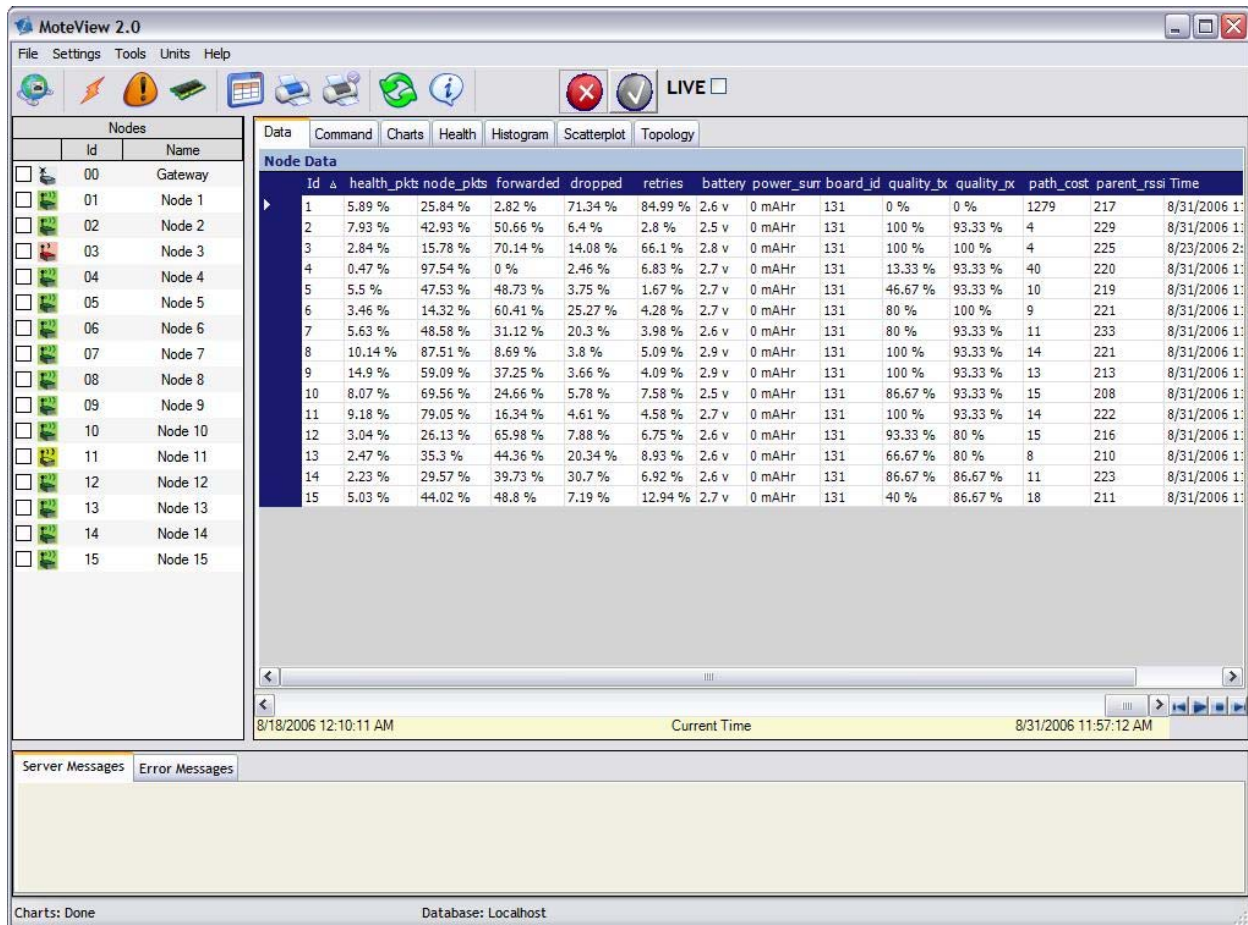


Figure 10-1. Sample Node Health Table

10.1 Node Health Packet Definitions

The `node_health` table contains the node ID, server timestamp and the following fields:

10.1.1 Health Packets

Health_pkts is the total number of node health and neighbor health packets that have originated at the node after the last reboot. This number is displayed in *raw units* (number of packets) and *% Packet* (ratio of health packets to node packets).

10.1.2 Node Packets

Node_pkts is the total number of packets that have originated at the node since the last reboot. This number includes all data, node health, neighbor health and route update packets.

10.1.3 Forwarded Packets

Forwarded is the total number of packets that the node has forwarded from other nodes. This number is displayed in *raw units* (number of packets) and *% Packet* (ratio of forwarded packets to node packets). A large number would indicate network congestion at this node.

10.1.4 Dropped Packets

Dropped is the total number of packets that the node has dropped. Packets are considered to have been dropped when 1 packets has been retransmitted 8 times without receiving the link-level acknowledgement. This number is displayed in *raw units* (number of packets) and *% Packet* (ratio of health packets to node packets).

10.1.5 Retries

Retries is the total number of retransmissions that the node had to make due to the fact that the node did receive a link-level acknowledgement.

10.1.6 Battery Voltage

Battery is a measure of the battery voltage available to the Mote.

10.1.7 Power Sum

This measurement is not currently supported by the *XMESH* firmware. This value will always be 0.

10.1.8 Board ID

Board_id indicates which sensor board is attached to the Mote.

10.1.9 Quality

Quality_tx and *quality_rx* are estimates of the link-level quality that takes packet collisions into account.

$$\text{Quality} = 1 / \text{Number of transmissions}$$

In *raw units* this is expressed as a number from 0-15 which corresponds to a quality of 0 – 100%.

Quality_tx is the link-level quality from the node to its parent.

Quality_rx is the link-level quality from the parent to a node.

10.1.10 Path Cost

Path Cost is an estimate of the number of transmissions required to send a packet from a node to its base. It takes into account the number of hops and the number of retransmissions per hop that is necessary for a node to send its packet to the base station.

$$\text{Path Cost} = 4 * \text{Estimated Transmission Number (ETX)}$$

$$\text{Path Cost} = 2^{10} / (\text{quality_tx} * \text{quality_rx})$$

The lowest possible cost is 4; this mean the node requires one transmission to send its packet to the base station.

This number is displayed as a *parent_path_cost* (path cost between a node's parent and the base station) and as a *node_path_cost* (path cost between a node and the base station).

10.1.11 Parent_RSSI

Parent_RSSI is the Received Signal Strength Indicator level of a parent's radio. MoteView displays the raw counts as signed 2's complement integers. To convert these counts to dBm use the following model

- MICAz Platform

Convert the unsigned raw counts (x) to a signed integer (y).

$$\text{If } x < 127; \quad y = x$$

$$\text{If } x > 127; \quad y = x - 256$$

Convert the signed counts (y) to RSSI in dBm, subtract -45 dBm

$$\text{RSSI} = y - 45 \text{ dBm}$$

Example:

$$\text{Raw Counts} = 217$$

$$\text{Convert to signed integer} \rightarrow 217 - 256 = -39$$

$$\text{Calculate RSSI (dBm)} \rightarrow -39 - 45 = \underline{-84 \text{ dBm}}$$

- MICA2 Platform

Calculate VRSSI = raw counts * battery voltage / 1024

$$P = -51.3 \text{ VRSSI} - 49.2 \text{ [dBm] at 433 MHz}$$

$$P = -50.0 \text{ VRSSI} - 45.5 \text{ [dBm] at 868 MHz}$$



Crossbow Technology, Inc.

4145 N. First Street

San Jose, CA 95134

Phone: 408.965.3300

Fax: 408.324.4840